SPREADSHEET GUIDE - PART TWO

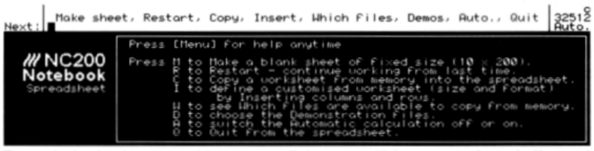
Complete TUTORIAL Guide

The following tutorial chapters work up from the most basic things you will need to know to start using the spreadsheet to some very advanced topics. As well as explaining the use of many of the commands that are covered in the complete command reference it also, by example, shows some uses to which the spreadsheet can be put.

Tutorial I - Basic Techniques

Creating a worksheet

This is the picture you will have on your screen after first starting the Notebook spreadsheet if you select the Start a new worksheet option.



```
Next:| Make sheet, Restart, Copy, Insert, Which Files, Demos, Auto., Quit |32512
                                                                          |Auto.
///NC200    Press [Menu] for help anytime
Notebook
Spreadsheet Press M to Make a blank sheet of fixed size (10 x 200).
                  R to Restart - continue working from last time.
                  C to Copy a worksheet from memory into the spreadsheet.
                  I to define a customised worksheet (size and format)
                      by Inserting columns and rows.
                  W to see Which Files are available to copy from memory.
                  D to choose the Demonstration Files.
                  A to switch the Automatic calculation off or on.
                  Q to Quit from the spreadsheet.
```

To get started as quickly as possible we will leave a description of building a sheet from scratch until later in the manual. For the time being just press [M] to select the Make sheet option. When you press [↵] a blank sheet with 10 columns and 200 lines will be defined. This is made up of cells that are 12 characters wide and have their format type set to General.

You will notice that it was only necessary to type [M] followed by [↵] to use the Make sheet option. You did not have to type the whole "Make sheet" command. The single key-stroke sequence used by the spreadsheet to symbolise each of the available commands should be extremely convenient to use but you must remember not to try type in the commands in full or some unexpected effects or error messages may appear. A similar form of prompting is used in a lot of computer program. They are normally referred to as menus, i.e. a list from which you make your selection.

When you are typing commands, if you make a mistake, just press the [Del] key and you will go back to the stage before your error. If you want to abandon the sequence entirely press the [Stop] key and you will be returned to the previous menu. No harm will be done.

The numbers that you now see running down the screen are the worksheet line numbers and the A..F above them are the worksheet column letters. The columns across and the rows down have divided the sheet into a grid. Each individual unit of this grid is called a CELL.

The cell cursor is shown as a cell printed in inverse text (currently the top cell in column A). This cell cursor marks the currently active cell which is where many of the commands you type in will take effect.

Each cell has a name, or co-ordinate reference, which you get by combining the column letter and the line number. At the moment the cursor is in column A and line 1 so the co-ordinate reference is A1. As this is the active cell you will see this co-ordinate displayed at the top left of the screen. The "Gen" below this tells you that the cell has a General format. This means that any number entered into it will be displayed in a general format - rather like you might see in a scientific calculator.

The active cell is the cell that is ready for you to enter data into it. Try this now by pressing the [=] key which signals to the spreadsheet that you want to type in some information. Type a few figures followed by [↵] and you will see your data appear in the active cell. In fact, if you want to enter numbers you don't even need to type [=] to start entry. Just start to type a number and the spreadsheet will automatically switch into entry mode. However, when you want to enter functions and formulae, you must start the entry with [=] .

Try moving the cell cursor about using the arrow keys and entering data into other cells. If you have had enough for one session use the [Stop] or Quit options to leave the spreadsheet. Either press the [Stop] key or type [Q] to select the Quit command. They have the same effect.

Tutorial II - Inserting and Deleting Rows and Columns

Now that you have had your first experience at using the sheet it is time to look more closely at the commands used for creating, and deleting parts of, a spreadsheet.

To get started quickly in the first tutorial we just used the Make sheet command which made a blank sheet of predetermined size and type. As an alternative you could have started from scratch using the Insert command to build up columns and lines of whatever size and type you desired.

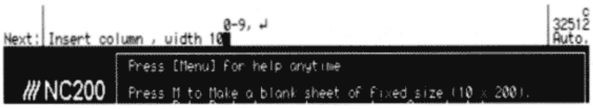
Having defined a sheet of a certain size it is also important to be able to remove unwanted parts of the grid. For this we use the Zap command.

The Insert and delete (Zap) commands act on rows or columns, either one at a time or in groups. If you still have a worksheet on screen from the previous section, clear anything you have done so far with [Z][A][↵][Y][↵] i.e. 'Zap All'. Note that the spreadsheet asks for confirmation before proceeding with this possibly disastrous command. You will return to the opening screen of the spreadsheet.

If you had already left the spreadsheet then select the **Start a new Worksheet** option. This time, instead of using [M] we will see how a sheet may be built up from scratch.

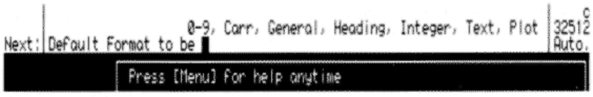
By choosing to build a worksheet in this way you have complete control over exactly how many columns and rows there are, how wide the columns are, what type of data they should contain and exactly how it should be displayed.

Now type [I][C]**10** for 'Insert Column of width 10':



```
                             0-9, ↵                               |C
Next:|Insert column , width 10                                    |32512
                                                                  |Auto.
///NC200  Press [Menu] for help anytime
          Press M to Make a blank sheet of fixed size (10 x 200).
```

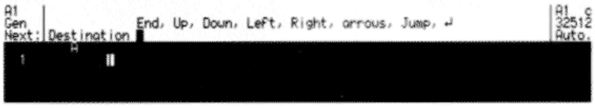
When you press [↵] you are then asked to give a default format for the column:



```
                  0-9, Carr, General, Heading, Integer, Text, Plot |C
Next:|Default Format to be                                         |32512
                                                                   |Auto.
           Press [Menu] for help anytime
```

Each new column you create will have its own default format, i.e. every single cell in the column is regarded as holding the same type of data (numbers or text) that is to be displayed in the same way unless you specify otherwise. You will learn about formats later, for now in response to the 'Default format to be' prompt type [G][↵] for General:

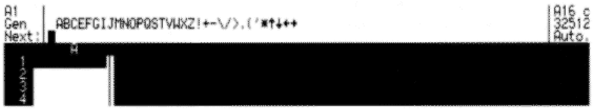
So far you only have a single cell so next try entering some extra lines with [I]**15**[L][↵] for Insert 15 Lines. You are then asked for a destination:



```
A1   |                                                   |A1   C
Gen  |        End, Up, Down, Left, Right, arrows, Jump, ↵|32512
Next:|Destination                                        |Auto.
        A
  1
```

As well as asking WHAT you want to add to the grid the spreadsheet also needs to know WHERE you want to put it so it prompted you for the Destination of the inserted lines.

If you just type [↵] when asked for this destination the new lines are put in front of the cursor. Your only alternative at this stage when only one line exists is [E][↵] for End in which case it will be put after the cursor. When you have a more complex sheet already defined there will be more possible options and you can insert new lines into the middle of the ones already defined. For now just type [↵] which will put the new lines above the current cell:

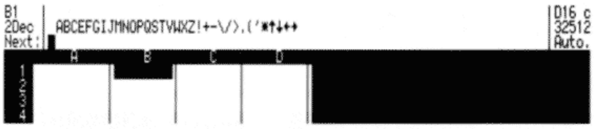


```
A1   |                                        |A16 C
Gen  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→      |32512
Next:|                                        |Auto.
       A
  1
  2
  3
  4
```

By inserting columns and rows in this way you can build the worksheet up to a possible maximum of 52 columns and 255 lines. The actual limitation on size will be dependent on the amount of free memory available in your Notebook. The spreadsheet is designed to use absolutely the minimum memory and so only those cells which are occupied with data use any memory at all. You can therefore start with as big a sheet as you like and fill in the detail later. If you have used other spreadsheets you will appreciate the difference.

You can use the Z key for Zap command to reduce the sheet by column, by line or as a whole (i.e. a clean start) with the ALL option. You will get more information about these later.

Now type the following command I3C8↵E↵2D↵. This will insert 3 columns with a width of 8 characters each to be placed at the end (i.e. to the right) of the current sheet. All the new cells will have a format of 2 decimal places.



```
B1    |                                                    |D16 c
2Dec  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→                  |32512
Next: |                                                    |Auto.
         A          B          C          D
1
2
3
4
```

Moving around the sheet

Try moving the cell cursor around the sheet you have created by typing ↓ for down. Note that the active cell co-ordinate at the top changes as the cursor moves down. Move the cursor back up again with ↑ for up.

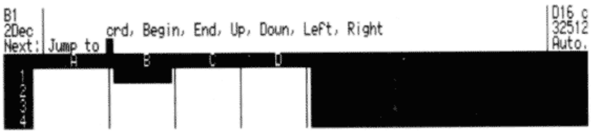
The arrow keys are probably the most natural keys to use to move the cursor but you will find that UDLR can also be used to go Up, Down, Left and Right. These options exist so that you can use the direction movement keys in macros. This is described later.

Alternatively, if you prefer to use direction movement keys laid out in a diamond shaped cluster you can use the WAZD keys - pressing them with Control. The positioning of these keys corresponds to the direction they move in.

The following were explained in the Getting Started part of this guide and are repeated here for your convenience:

↑, U, Control-W	cursor up
↓, D, Control-Z	cursor down
←, L, Control-A	cursor left
→, R, Control-D	cursor right
⇧↑ or -	Move up a page
⇧↓ or +	Move down a page
⇧←	jump to left edge on current line
⇧→	jump to right edge on current line
Control↑	jump to top cell in current column
Control↓	jump to bottom cell in current column
Control←	jump to cell at top left of defined worksheet
Control→	jump to cell at bottom right of defined worksheet.

If you have to move the cursor to a specific cell you may find it easiest to use the Jump command. This command goes directly to the co-ordinate that you specify. Type J and you will see:



```
B1    |                                                    |D16 c
2Dec  |        crd, Begin, End, Up, Down, Left, Right      |32512
Next: | Jump to                                            |Auto.
         A          B          C          D
1
2
3
4
```

The "crd" in this prompt means that you can specify a co-ordinate as the destination for the jump. Type **D6**↵ to move to the cell at the bottom right of the sheet. If you just received the message "No such column" this is because you typed Jd6, rather than JD6 - remember that the first 26 columns are labelled with the upper case letters so you must use D, not d.

There are six other possible jump destinations - Begin and End, which refer to long jumps to the very beginning (top left) and very end (bottom right) of your worksheet. Right and Left go to the very end and beginning of the current line and Up and Down go to the top or bottom of the current column. Some of the ⇧ and Control + arrow key commands actually call these forms of the Jump command and you may find it easier to use those but the Jump command has the advantage that it always reminds you what the possible jumps are.

Now use JB↵ to move the cursor to column A, line 1 (that is, cell A1) and press . to start entering a number. Type **37**, when you hit ↵ you will see the number you entered displayed in the active cell. Now move the cursor right into one of the three columns that were created with a format set to 2 decimal places. Enter the same number in one of these. Once again you see the number displayed but there will be 2 places after the decimal point shown in the cell. This is because columns B to D have their display format set to two decimal places. Notice that the status information at the start of the second display line says "2Dec". In column A it was "Gen" which shows that it has a general display format.

A note on destinations

When using Insert you will notice that columns can be inserted on either side of the current column using either the left or right arrow keys. Other options are at the beginning and end of the sheet. When you are inserting any lines or columns you will be prompted with this range of options for the Destination:

```
End, Up, Down, Left, Right, arrows, Jump, ↵
```

Of these options UDLR, arrows and J all serve to move the current cursor position to the point where you want the new line/column to be inserted. When you have the cursor positioned where you want it in the sheet use the ↵ key to make the insertion.

A new column will be inserted to the left of the column containing the cursor when ↵ is pressed. A new line will be inserted above the line containing the cursor when ↵ is pressed.

End is a special command that simply tells the spreadsheet to tack on a new line or column at the End borders of the current sheet i.e.

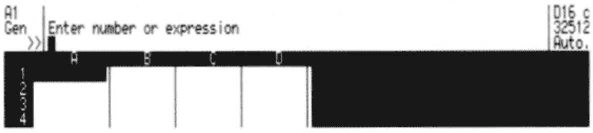
Columns are added to the right of the sheet, lines are added to the bottom.

When inserting a line or column, the cursor is positioned in the new line or column ready for the insertion of data.

More on Entering Data

Clear any numbers you have entered so far by typing BA↵Del→↵. This uses a new command, Blank, to clear data from occupied cells but leave the empty cells behind. This is similar to the Zap command except that Zap also removes the empty cells. If the cursor isn't there already, type JB↵ to jump to the beginning of the sheet (cell A1).

As you have seen, when you type . the screen will change to show this prompt:



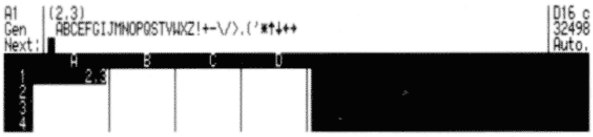
```
A1    |                                                    |D16 c
Gen   | Enter number or expression                         |32512
   >> |                                                    |Auto.
         A          B          C          D
1
2
3
4
```

This is called entry mode and as you have already seen, the spreadsheet is now ready to accept some form of data. This can either be text, a number or an expression. These are the three types of entry that are possible. The possibilities open to you are determined by the broad choice of format type you have made, i.e. between a format suitable for text or one suitable for numbers.

You can use > instead of . to start entering data if you prefer. The two are exactly equivalent. There are other keys that can be used to start data entry that are described in Tutorial III.

The amount of data that can be entered into a cell is limited by the width of the Entry line - 67 characters.

The worksheet we have defined so far has only used formats suitable for numeric data - General and 2 decimal places. The prompt you receive will reflect this. Try it now, enter a number **2.3** and press ↵. If you make a mistake, press the ←Del key and the cue will backspace removing the last character you typed. Your screen will look like this:

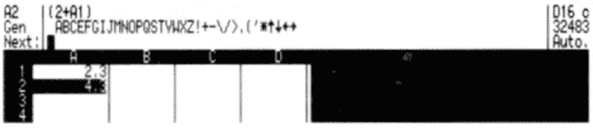


```
A1    |(2.3)                                    |D16 c
Gen   | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→        |32498
Next: |                                         |Auto.
       A        B        C        D
 1      2.3
 2
 3
 4
```

Note that the contents line at the top now has your entry enclosed in parentheses - (2.3). The brackets are an indication that the entry is a number or expression. You will see later that if it had been an expression the contents line would show exactly what you had typed but the worksheet display would only show the calculated result.

Note also that the memory, shown by the figure at the end of the second display line, has gone down a little.

You are next going to see the power of the spreadsheet. Move the cell cursor down with the [↓] key and press [>] again. This time enter **2+A1**[↵] which is an expression meaning '2 plus the current value at co-ordinate A1'. Remember to type the A in upper case. You will now see:



```
A2    |(2+A1)                                   |D16 c
Gen   | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→        |32483
Next: |                                         |Auto.
       A        B        C        D
 1      2.3
 2      4.3
 3
 4
```

Note that the RESULT has been calculated and shown in the display.

Go back to A1, select entry mode again and enter a new value of **3.7**. When you press [↵] you will note that A2 changes at the same time. You have managed to enter an expression that uses a value from another cell and this expression works, however, you change that cell value. The same principle can be extended to operate over the most complex of worksheets and formulae.

Note that although the sheet display shows the RESULTS of the formula you enter, the contents line still shows the expression that you entered in the brackets. Move the cursor back to A2 and notice that although the cell in the worksheet is displaying 5.7, the contents line shows (2+A1).

Note also how easy it is to alter an existing cell entry by just typing in a new value.

One thing you may find useful is that if you use the arrow keys to move the cursor while in the process of entering text or expressions then it will finish off that entry and move the cursor to the cell in the direction you have specified. This cell will be set up ready for you to make your next entry. The cell format will be set up to be exactly the same as the previous entry.

The use of arrows in this way can save much time with long lists as the one key stroke is equivalent to [↵], cursor movement and [>] for entry. Obviously the [L][R][U][D] keys cannot be used in this case since the spreadsheet cannot tell whether or not you are intending to enter some text, a cell reference or a function name. You can, however, use [Control]-[W],[A],[D],[Z] if you prefer.

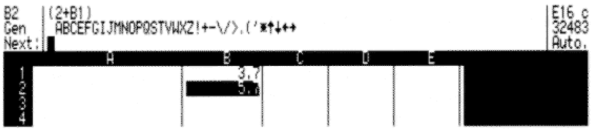
By changing a single value a huge number of other cells that are dependent on it will all change at the same time. In this way the Notebook spreadsheet gives you the power to test the effect of different values on an answer and so do 'What if?' assessments of a situation.

Entering Text

Now you are going to create another column and insert some text instead of a number. With the cursor still in column A, type [I][C]**20**[↵][↵][T][L][↵]. Some of these commands we have met before. The TL command is the default format for the column we have created. T stands for Text which tells the spreadsheet what type of information to expect. L stands for Left justified which just means print the text from the left hand edge of the active cell.

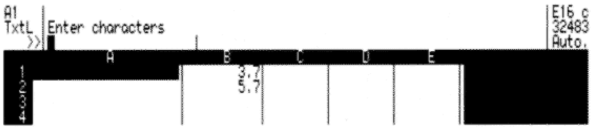
Column A is now set up for you to put in the text. Note that the original column has been shifted to the right and labelled B. This again shows how easy it is to alter the dimensions of your worksheet grid at any time.

Before you go any further, use the [→] key to move the cursor to location B2 which contains the expression we have already typed in. If you look at the top line of the screen you will see that the A1 which you have typed in has now changed to B1 to keep track of the effect of entering another column. You can rest assured that your expression will still work the way you intended it to.



```
B2    |(2+B1)                                   |E16 c
Gen   | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→        |32483
Next: |                                         |Auto.
            A           B      C      D      E
 1                     3.7
 2                     5.7
 3
 4
```

Go back to A1 using the [←] key for left and the [↑] key for up. Type [>] and you will see this:

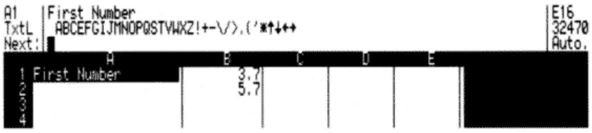


```
A1    |                                         |E16 c
TxtL  |Enter characters                         |32483
  >>  |                                         |Auto.
            A           B      C      D      E
 1                     3.7
 2                     5.7
 3
 4
```

This time you are asked to enter characters because the cell has a text, rather than numeric, format. You may have noticed that the start of the second line in the status area of the screen is now showing TxtL which reminds you that the column's default format is Text Left justified.

The Entry Line will display a marker (the vertical line just above the column B label) that reminds you of the current display width for the cell, but this can be exceeded - the actual amount of data that can be entered is only limited by the width of the Entry line.

Make sure Caps Lock is switched off then enter some text, type **First number**[↵]. You should see:



```
A1    |First Number                             |E16
TxtL  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→        |32470
Next: |                                         |Auto.
            A           B      C      D      E
 1 First Number        3.7
 2                     5.7
 3
 4
```

Recap:

Now look at this list to remind yourself of some of the features you have seen so far.

[M]	is used from the opening screen to quickly make a blank sheet.
[I]	is used to insert columns and lines. You have to say how wide any new columns will be and what type of format they should have.
[U] or [↑]	moves the cell cursor up.
[D] or [↓]	moves the cell cursor down.
[L] or [←]	moves the cell cursor left.
[R] or [→]	moves the cell cursor right
[J]	is used to jump to a particular cell.
[>]	allows you to enter a number, expression or text.
[↵]	is used to accept and complete a command.
[Stop]	press this key to abandon the current operation, without harm.
[←Del]	use this key to backspace and correct mistakes while entering data or commands.
A1,A2	are cell co-ordinates and may be used if they are numbers in an expression.

You have now tried both expression entry and text entry. The choice of whether text or number is to be entered depends on what format the cell was given when it was created.

You have also seen that the same number entered into cells with different formats is displayed differently.

Tutorial III - More advanced techniques

Understanding the difference between text and numbers

You will have seen already that the data that has been entered can be displayed in a variety of different ways, as defined by the current format.

There are about a dozen built in formats that can be used but they divide into two broad types, some relating to text and some relating to numeric data. It is possible to switch formats as long as the cell is empty (using the Format command) or the data within a given cell can conform to the new type (using the New Format command).

For example you can switch a number between Integer format, which only displays the whole number part of a value, to a Financial format, which displays data correct to two decimal places. It does not make sense to try and switch Text to an Integer format for example and the spreadsheet will catch any attempt to do so during the entry process. Although you might think that it should be possible to switch a number or expression to a text format this is not possible.

Text data cannot be processed to a numeric value; you could enter a mathematical expression into a a cell that is expecting text data but you would not get a numerical answer. This is an important distinction for you to grasp. If for example you enter 23+23 into a numerical cell the spreadsheet will realise that it is expected to work out the answer and display 46. Entering exactly the same thing into a text cell will simply cause "23+23" to be displayed in the cell.

Text is used for headings, explanatory labels or for information in a database such as names and addresses. The Notebook spreadsheet assumes that you may want to enter numbers and formulae into these cells as part of an explanation of the surrounding sheet so no attempt is made to calculate using these values.

Text data can NOT be entered into a cell that has been set up with a numeric format.

It is possible to use the EDIT command (described later) to change data to the wrong format but you will find that the error will be pointed out to you as soon as you leave the Edit mode. Pressing the ←Del key will return you to edit mode to correct your mistakes.

Data can be assigned to an incorrectly formatted cell if it has been loaded in from a stored file onto an existing sheet. However, any attempt at performing a calculation on the data will throw up the error.

Using Ranges

Certain of the Notebook spreadsheet's built in functions and expressions work on a specified range or column of the data. For example to AVERAGE some of the data you would specify the range thus

```
AVERAGE (B1...B10)
```

When you type this in you only actually type a single `.` and it is automatically expanded to `...`. Any blank cells are ignored in the above calculation. This saves you from having to create unnecessarily complicated expressions in order to encompass all of the data required. However, if an occupied, text format cell is included in the range you will see the error **`Text reference in an expression`**.

Formats: changing the way the data is displayed

You will remember that the format does not affect the actual data value that a cell holds, just the way it is displayed on screen. You can therefore change a numeric cell from decimal format to Integer and back again without losing any precision of the data. Whilst you are moving around the sheet the contents line at the top of the screen will always show the data or expression of the current cell exactly as it was originally entered, regardless of the current format.

So far you have only seen the General and 2 decimal place format in expressions and TL (Text Left justified) for text.

If you are still unclear about what is meant by changing the type of display format, consider this example. If you have a number such as 2 there are several ways you could write it down such as 2, 2.0, 2.000, 0.2E1.

All are perfectly valid but not all are as you would want for presentation in a report, or for immediate legibility. If it was referring to a whole number of items you would want 2. If it was the amount of money you would want 2.00. If it was a laboratory test result you may want to infer a precision to the nearest 0.1 by using 2.0.

Just in case you don't know 0.2E1 is known as "exponential" or "scientific" notation and means 0.2 x 10^1. This is just another way of entering and displaying numbers. This is often used when representing particularly big or small numbers. It is far easier to work with 0.3825E14 than 38250000000000 !

You can produce any of the above layouts using the various formats available with the Notebook spreadsheet. The full list of available numerical format types is:

Gen	General, rather like a scientific calculator
#Fin	Finance, balance sheet format (# is number of places to shift by)
#Dec	Decimal places specified (# is number of places to be displayed)
#Exp	Exponent scientific notation (# is number of significant figures)
Int	Integer, nearest whole number
Plot	Plot format, horizontal bar graph

Text formats can be:

TxtL	text left justified
TxtR	text right justified
Head	heading

Careful use of the format options can help to produce some quite sophisticated printouts. Notice that as you move around the sheet the start of the second display line always shows the format type of the current cell using a 3 or 4 letter code as described in the list of formats above.

There is also a format choice of Carriage Return that can be assigned to individual Cells or ranges. This is a special format used to control printout in such a way that address labels can be produced. No data can actually be entered in cells that have this format. When printed, cells with this format cause a new line to be started on the printer.

If you use Insert to create new columns you are asked to give the cells you are creating a default column format type. If you select any of the numeric formats then subsequent use of >. to enter data will ask you to enter a number or mathematical expression. If, on the other hand, the cells were created with a default column text format then using >. will ask you to enter characters.

The >. command starts data input using the current default column format. However, you may wish to over-ride this for entering just single pieces of data. There are several ways to do this.

If you start data entry by typing the F (Format) command you will first be asked the exact format to be used for that particular cell and will then be invited to enter the corresponding data.

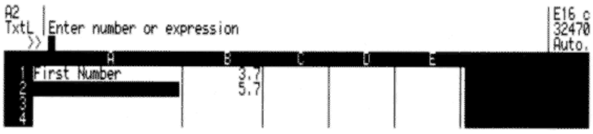
If, instead of `.` you start data entry by typing `(` this always means that you want to enter a number/expression, rather than a piece of text, even if the cell was created with a text format. In this case you are not asked to specify the exact format to be used. Instead, the Default Global Numeric Format is used. When you first start the spreadsheet this is set to be General format. You can use the NG (New Global format) command to change the default global format. This is explained later.

If, on the other hand, you start data entry by typing a quote character (either `'` or `"`) the spreadsheet will assume you want to enter a piece of text even if the current cell was initially created to have a numeric format. This uses the Default Global Text Format which is initially set to Text Left justified.

You may also change the default format used in a column when the >. command is used. This is done with the ND (New Default format) command. If some cells are already occupied and have associated formats they will not change but any data subsequently entered into blank cells in the column using [>.] will take on the newly set default column format.

More on Numeric formats

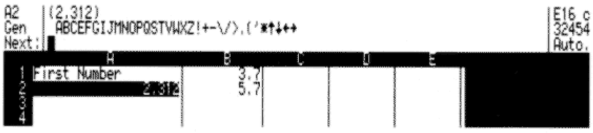
You can now experiment with some different formats. Go to cell A2 (which is in a column with Text Left format) and type [F][G][↵] which means 'make this cell General Format and start data entry'.



```
A2   |                                          |E16 c
TxtL |Enter number or expression                |32470
  >> |                                          |Auto.
          A           B        C       D       E
 1 First Number      3.7
 2                   5.7
 3
 4
```

Because you have chosen a numeric format the type of prompt you will receive will reflect this i.e. Enter number or expression.

Type **2.312**[↵]:

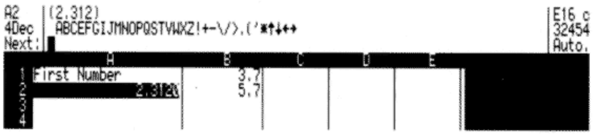


```
A2   |(2.312)                                   |E16 c
Gen  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→         |32454
Next:|                                          |Auto.
          A           B        C       D       E
 1 First Number      3.7
 2            2.312  5.7
 3
 4
```

You will see that your value of 2.312 is displayed just as you typed it in.

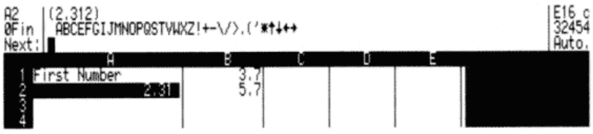
The [F] for Format command should only be used on cells where you wish to change both the format and the data contained. It is also possible to change just the format of an existing entry without having to re-enter the data. Without moving the cell cursor type the following and watch the changes.

Type [N][F]**4**[D][↵] for 'New Format to be 4 places Decimal':



```
A2   |(2.312)                                   |E16 c
4Dec | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→         |32454
Next:|                                          |Auto.
          A           B        C       D       E
 1 First Number      3.7
 2           2.3120  5.7
 3
 4
```

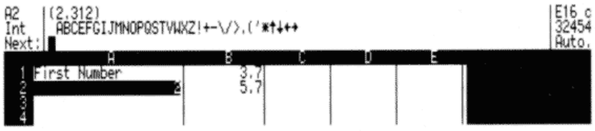
Type [N][F]**0**[F][↵] for 'New Format, no shifted places, Financial':



```
A2   |(2.312)                                   |E16 c
0Fin | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→         |32454
Next:|                                          |Auto.
          A           B        C       D       E
 1 First Number      3.7
 2             2.31  5.7
 3
 4
```

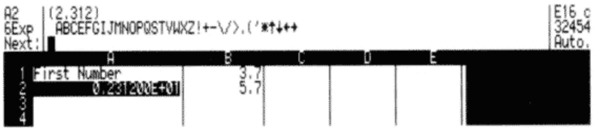
The Financial format always shows 2 decimal places and will put commas into the numbers as they get larger. Negatives will be enclosed in brackets. You can have a financial value shifted (usually by 3 or 6 places) when you want thousands or millions, etc. to be shown in a small number of digits.

Type [N][F][I][↵] for 'New Format Integer':



```
A2   |(2.312)                                   |E16 c
Int  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→         |32454
Next:|                                          |Auto.
          A           B        C       D       E
 1 First Number      3.7
 2                2  5.7
 3
 4
```

Now try [N][F]**6**[E][↵] 'New Format 6 significant figures Exponent' notation:



```
A2   |(2.312)                                   |E16 c
6Exp | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→         |32454
Next:|                                          |Auto.
          A           B        C       D       E
 1 First Number      3.7
 2     0.231200E+01  5.7
 3
 4
```

It is important to remember that however you display a value there is no change in the way the number is stored by the spreadsheet. Formatting does not change the value that you have entered or calculated, just how it is laid out on the screen or printed on paper. The contents line at the top of the screen always shows what is actually held in the cell.

Note that cells themselves do not have a particular format, it is only once a piece of data is entered that it has a display format associated and stored with it. It may get this format either because you specifically ask for it using the Format command or it may come from the default column format if you use the [>.] command. The default column format is that which you specified when the column was inserted or when you used the ND command. Entered data may take on the global numeric or global text format if you start entry of the data with **(** or **"** respectively.

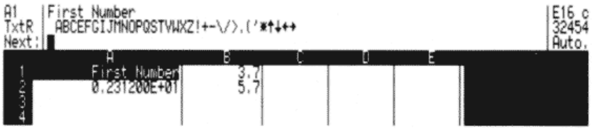
Because a format is associated with a piece of data rather than a cell you cannot use the NF (New Format) command on a cell that has not yet had a piece of data entered into it. You also find that if you move or copy data the data takes its format with it to the new location.

Text Formats

There are two normal types of text display. Justified in the column width is the first type. This has two subdivisions - you can specify that you want the text to be left justified (it hugs the left hand side of the column) or right justified (it hugs the right hand side of the column).

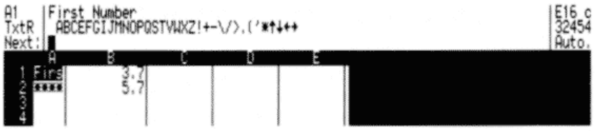
The second type of text entry is Heading. This type will overwrite the column(s) to the right if there is not enough space to display all of the information in the single cell. As the name suggests this is most useful when entering explanatory titles or headings that you want to be displayed regardless of any changes in the various column widths on screen.

The entry in cell A1, **First Number**, is currently shown in text left justified format as this is the default format of column A. Move the cursor to cell A1 then type [N][F][T][R][↵], which stands for 'New Format Text Right justified' to see the effect of changing the format to right justification without losing the data.



```
A1   |First Number                              |E16 c
TxtR | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→         |32454
Next:|                                          |Auto.
          A           B        C       D       E
 1     First Number  3.7
 2     0.231200E+01  5.7
 3
 4
```

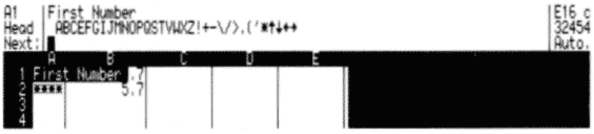
Before investigating Heading format you are going to change the column width. Type [N][W]**4** [↵] which stands for 'New Width 4'.



```
A1   |First Number                              |E16 c
TxtR | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→         |32454
Next:|                                          |Auto.
      A     B        C       D       E
 1 Firs   3.7
 2 ****   5.7
 3
 4
```

You should note that only the start of the word "first" can be displayed in the available space now that we have made the column much narrower. The whole entry is still there in the worksheet memory as you can see by looking at the contents line at the top.

If you type [N][F][H][↵] for 'New Format Heading' you will see:



```
A1   |First Number                              |E16 c
Head | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→         |32454
Next:|                                          |Auto.
      A     B        C       D       E
 1 First Number.7
 2 ****   5.7
 3
 4
```

The whole of **First Number** is again displayed and part of it overwrites cell B2.

Recap:

You have learnt that data falls into two broad subdivisions, text and numeric. We have then looked at some of the format subdivisions within those two types that control the way the data is displayed on screen.

You have seen that it is only what is displayed that changes. The same value that you originally entered is always stored and used in calculations - however you choose to display the result.

You have seen that when entering data into a cell one of several different formats may be used. The column format (set when the column was created) is used if you start data input with [>.]. A specific format of your choice is used if you start data input with the [F] command. A global numeric format is used if you start data input with **(** and a global text format is used if you start data input with **'** or **"**.

Tutorial IV

More on changing column widths and formats

You have already used the New command several times to change the column width and to change the format of an entry. It can also be used to change the default column format. The default column format is the format that will always be used in your current column if you do not declare a particular one before each entry. You enter data using this type of format by starting with the [>.] command.

The last item that may be changed is the global format. Despite the singular name this is a pre-set pair of formats, one for text and the other for numeric values, that may be used anywhere on your worksheet and which are quicker to call up than by explicitly stating which format type you require - they are time saving devices. You can call these formats instantly by using the **(** command for numerical expressions and the ' or " command for text. At start-up these formats are set to General for numerical expressions and Text Left Justified.

If the vast majority of your worksheet is likely to use one format then it will be worth your while redefining the Global formats.

If you type [N][G] and then a numeric format you will change the default global numeric format, while, if you type [N][G] followed by one of the three text formats you will set the default global text format. So the same NG command may be used to set two different global default formats.

The complete set of New commands is:

NW# New column width
(# is number of spaces to set width to)

NF New format for a cell already containing data

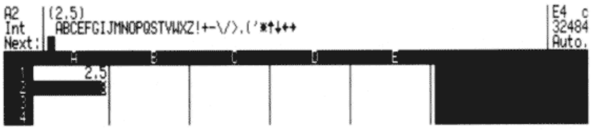
ND New default column format

NG New global format

You haven't seen the Global format in operation yet and so here are a couple of examples. Type [Z][A][↵][Y][↵] to clear any sheet you have loaded, alternatively, start the spreadsheet with the "start a new worksheet" option. Then type [I]**5**[C]**10**[↵][G][↵] then [I]**3**[L][↵][↵] to set up a small sheet of general format cells.

Now type [N][G][I][↵] to change the global numeric format to integer.

Type [>.]**2.5**[↵] to enter the value into cell A1. It is displayed as '2.5' because it is in general format. Now move to A2 then type **(2.5**[↵] and the number is displayed as '3' because it is in integer format.



```
A2    |(2,5)                                    |E4  c
Int   | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→       |32484
Next: |                                         |Auto.
         A        B        C        D        E
  1       2.5
  2         3
  3
  4
```

The operation is very similar if you want to put text in a column that is set up for a number. This time use the quote to start text entry using the Global text format.

You can use the Global format entry method anywhere on your worksheet. You can, of course, also change the Global text format using the New command.

Remember that you cannot use the New Format command for changing between text and numeric formats. The NF command will only work on an occupied cell because only pieces of data that have been entered actually have a format associated with them which can be changed.

Even if you have managed to get the wrong type of data into a cell, using the Edit command or by loading a file in, you still cannot change the Format between text and numeric in order to suit the data. You will have to rewrite what is there AFTER changing the format.

Recap:

We have seen how the global format can be used to quickly set up cells for a certain data type. This allows the existing column format setting to be over ridden.

Editing

We have seen already how easy it is to replace the contents of a cell with something different.

For most operations you will find that this is the most convenient method to use for making changes. However, sometimes you may wish to make only minor modifications to an existing entry and for long and complex expressions it would be more convenient just to edit the existing information.

If you decide to change an entry without re-entering it you use the spreadsheet's cell editing commands. You just position the cell cursor on the cell you would like to change and type [E][↵]. The current contents of the cell as you originally entered it are displayed on the entry line and you can then amend it as necessary and press [↵] when finished.

While editing you can type new characters which will be inserted before the cue. You can move the cue along the existing line using the [Word ←] and [Calc →] keys. Existing characters can be deleted using either of the [Del→] or [←Del] keys.

At any time before you finally press [↵] you can press the [Stop] key. This abandons the changes you have made and retrieves the entry in the form it was before you entered the edit mode.

The important thing to remember when you edit a piece of data in this way (rather than entering it from scratch) is that the automatic error checking is turned off so you can change the value in a cell to something quite ridiculous and it is only when you finally press [↵] to leave the editor that the error will be detected.

Tutorial V

Removing data from the sheet - Blanking and Zapping

Let us recap on the Zap command. So far you have only used the sequence [Z][A][↵][Y][↵] to remove (Zap) the whole sheet. However, this command also works on specified lines or columns, either individually or collectively. It not only removes the data from the cells in question but also reduces the size of the sheet. The lines or columns that are removed need not be on the outer edges of the sheet.

You are prompted to specify which rows and columns to remove - the remaining cell co-ordinates will automatically adjust themselves to reflect the change.

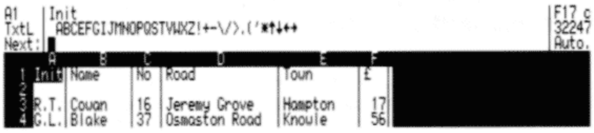
Cells cannot be zapped if there are other formulae that depend on the data held within. If you try to do so you will be offered the chance to use the BLANK option instead.

If you are sure you want to Zap a column or line that has data dependent on it, as above, you must first find and zap or blank all of the cell formulae that depend on the value held in the target cells for their calculation. This is more simply done than it may first seem because the spreadsheet will tell you which are the dependent cells in turn.

Although the Zap command also offers the option of Entry or Block, if you select one of these it will only blank them. It would be impossible to completely remove a single cell (or a block) as this would leave a hole in the middle of the sheet.

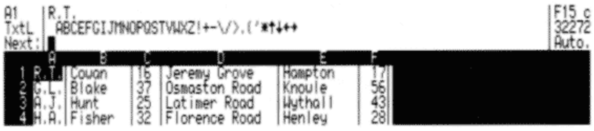
As an example you will see how to use some of the other forms of zap command in more detail.

Zap any sheet you are working on with the command Z A ↵ Y ↵ then copy in the demo file called ADLIST by typing D **ADLIST**↵.



```
A1    |Init                                         |F17 c
TxtL  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→           |32247
Next: |                                             |Auto.
     A     B       C    D               E       F
  1 Init|Name     |No |Road            |Town    |£ |
  2
  3 R.T.|Cowan    |16 |Jeremy Grove    |Hampton | 17|
  4 G.L.|Blake    |37 |Osmaston Road   |Knowle  | 56|
```

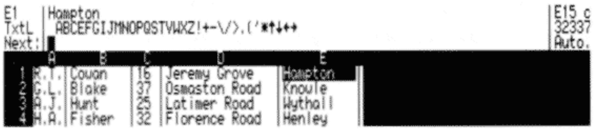
Suppose you do not want the top two lines which are just labels. You could remove these by positioning the cursor anywhere in line 1 and typing Z 2 L ↵. Those lines are removed and the original line 3 is moved up to line 1.



```
A1    |R.T.                                         |F15 c
TxtL  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→           |32272
Next: |                                             |Auto.
     A     B       C    D               E       F
  1 R.T.|Cowan    |16 |Jeremy Grove    |Hampton | 17|
  2 G.L.|Blake    |37 |Osmaston Road   |Knowle  | 56|
  3 A.J.|Hunt     |25 |Latimer Road    |Wythall | 43|
  4 H.A.|Fisher   |32 |Florence Road   |Henley  | 28|
```

This example also demonstrated a feature that is common to many of the spreadsheet commands (zap, blank, copy, move, insert) that you can normally specify a number to say that you wanted it repeated for several columns or lines. This is normally indicated when the prompt offers the option **0-9**.

Suppose you wanted to get rid of column F which just contains some numeric values. Move the cursor to column F by pressing ^ →. You can now remove the column with Z C ↵.



```
E1    |Hampton                                      |E15 c
TxtL  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→           |32337
Next: |                                             |Auto.
     A     B       C    D               E
  1 R.T.|Cowan    |16 |Jeremy Grove    |Hampton |
  2 G.L.|Blake    |37 |Osmaston Road   |Knowle  |
  3 A.J.|Hunt     |25 |Latimer Road    |Wythall |
  4 H.A.|Fisher   |32 |Florence Road   |Henley  |
```

Blanking

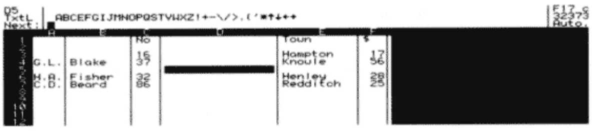
You can use the Blank command to clear all, or parts, of the worksheet. In contrast to the Zap command you will not change the size of the sheet or any of the default settings with this option.

However, individually set commands such as formats will revert to the default column ones.

As an example you will see how to use the block option in more detail. Get back to the opening menu by zapping the sheet you were working on with the command Z A ↵ Y ↵ then copy in the demo file called ADLIST again by typing D **ADLIST**↵

Type B B **A1**. **B3**↵ which stands for 'Blank Block in the range A1 to B3'. Note that you are allowed to type a single full stop to specify a range but the Notebook spreadsheet automatically expands this to three for clarity. Remember that the cell co-ordinates must be A1 and B3 in upper case. If you type a1.b3 the spreadsheet would think you meant columns 27 and 28 rather than 1 and 2.

Now blank column D by moving the cursor to any cell in column D and using the command B C ↵. Move the cursor to line 4 and type B L ↵ to blank that line.



```
D5    |                                             |F17 c
TxtL  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→           |32373
Next: |                                             |Auto.
     A     B       C    D               E       F
  1                |No |               |Town    |$ |
  2
  3                |16 |               |Hampton | 17|
  4 G.L.|Blake     |37 |               |Knowle  | 56|
  5
  6 H.A.|Fisher    |32 |               |Henley  | 28|
  7 C.D.|Beard     |86 |               |Redditch| 25|
  8
  9
 10
 11
 12
```

You will never be allowed to inadvertently destroy the sense of the sheet with the Blank command. If you have an entry like 2*B3 and try to blank B3 the spreadsheet will check and warn you. However, you will be allowed to confirm that you wish to blank the cell. A row of '???' will then appear in that cell at every recalculation to signal to you that some data is missing from the sheet.

Depending on the formulae that refer to the blanked cell you may also trigger other errors such as 'Division by zero'.

Recap:

You have learned how to reduce the size of the sheet using Zap and how to erase entries using Blank.

Tutorial VI - Expressions

Understanding expressions - the heart of the sheet

We have already looked in brief at entering expressions in the first part of the Tutorial where we just used '2+A1' to add 2 to A1. You will not be surprised to find that there is very much more to expression formulae than that. However, just take it slowly experimenting with each new feature and finding those that suit your particular needs.

A worksheet that contains just numbers and text can be no more useful than a piece of paper; you can enter values that have relationships with each other, but the spreadsheet, like the paper, can't guess what these relationships are or do the work of the working out for you. To really exploit the power of a spreadsheet you must get used to entering the relationships between the data, not the results of the relationships, and letting the computer do the calculating.

For example, in a financial report, we may have an entry for yearly income, one for costs and one for profit. We understand that the value that represents profits is worked out as Income minus Costs.

When using the spreadsheet we should enter only the data for income (in cell A1 for example) and for costs (e.g. cell B1), in the third cell we then enter the relationship between these two that represents profits in the form of a formula (A1-B1). The technical term for this formula is an expression.

The power of the spreadsheet lies in the fact that you can enter extremely complex formulae and inter-relationships between the cells, but as soon as you make a change to one entry the full repercussions of it can be calculated and the entire sheet almost instantly updated.

Expressions are at the heart of the spreadsheet, for with them you can do business calculations, as well as scientific and engineering mathematics. An expression is defined as any entry that can be used to calculate a value.

What are the features of an expression? We have said you may think of it as a mathematical formula. 2+2 is an expression which should equal 4. 2+A1 is also an expression which as you have already seen is equal to 2 plus whatever is the value of A1. You can use any of the following mathematical signs, known correctly as OPERATORS:

+	plus
-	minus
*	multiply
/	divide
^	to the power of
%	percent

The + and - signs you will, of course, be familiar with, but if you are new to computing the symbols *, / and ^ may be unfamiliar. You will always have to use the ^ symbol when you want to get 'to the power of' because super scripts are not available, so for example, instead of using 5^3 you use 5^3 which gives the result 125 (=5*5*5). The % symbol is not often used in calculations outside of spreadsheets. It means 'percent' such that 5%20 is 5 percent of 20, which equals 1.

You will now see how to enter a formula into a cell. You may already be wondering how to enter the formula on one line when usually they take up 2 or more if written out longhand. This is simply done by splitting the formula into parts and enclosing these in brackets ( ). To show that one part is divided by another we put / between them, so:

$\frac{2+A1}{5+A2}$ becomes (2+A1)/(5+A2)

This is a common computing convention and will be familiar to anyone who has used a programming language such as BASIC. If necessary you can have several layers of brackets to avoid ambiguities. If the spreadsheet finds a reference to a cell that does not have a value it will use 0 and the blank cell will show a line of question marks to show that a value is expected. This is a useful feature that allows you to use a range of cells in some functions without worrying whether they all have the correct form of data.

The Notebook spreadsheet works out expressions in normal algebra. This is the algebra you use in hand calculations. Some people with experience in computing may be relieved to hear that the back to front entry method (reverse polish notation) is not used.

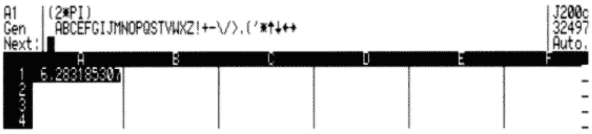
The spreadsheet understands the correct order of precedence of the operators during calculation. In general make up formulae as you would for pencil and paper, convert them to one line form and enter them and the result will come out as you intended.

On many spreadsheets there is no built in order of precedence; 1+2/5 would be worked out to be 0.6, by adding the 1 to 2 before dividing by 5. This is not the correct result, the real answer is 1.4 - that is, divide the 2 by 5 BEFORE adding it to 1 because the division operator has precedence over the addition operator. In spreadsheets that do not have a built in order of precedence you have to take positive action to force a correct result by entering 1+(2/5). It is unfortunately easy to get unexpected results, but you will have no such problems when using the Notebook spreadsheet. The order of precedence of calculation is:

()
%
^
* and /
+ and -
>, < and =   these are logical operators.
], [and |

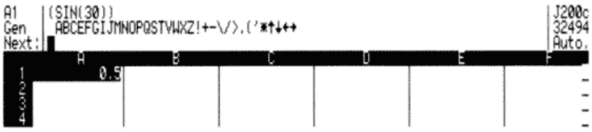
The Notebook spreadsheet starts the calculation with the innermost brackets. Within the brackets it works out the part of the expression starting with the highest precedence operator. Where there are two operators of equal precedence the one on the left is done first. The next bracket is then done and so on. Do not worry about the details - just make up your expressions in your usual way.

The other item you can use in an expression is a function. This is a built in formula that can be called by a simple name and which will return a value to your expression. The simplest functions are ***e*** and **PI** which, when used in calculations, give the values 2.71828182845905 and 3.14159265358979 without having to enter them. PI is the ratio of a circle's circumference to its diameter. *e* is the base of natural logarithms.



```
A1    |(2*PI)                              |J200c
Gen   | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→  |32497
Next: |                                    |Auto.
          A        B        C        D        E        F
 1  6.283185307
 2
 3
 4
```

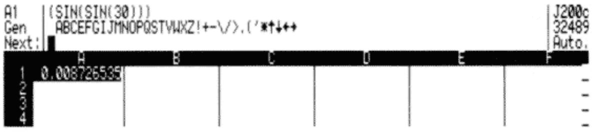
There are also some functions into which you must first insert a value. An example is **SIN(*n*)** which means 'work out the sine of *n* degrees'. Note that the value you enter into the function must be enclosed in brackets:



```
A1    |(SIN(30))                           |J200c
Gen   | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→  |32494
Next: |                                    |Auto.
          A        B        C        D        E        F
 1           0.5
 2
 3
 4
```

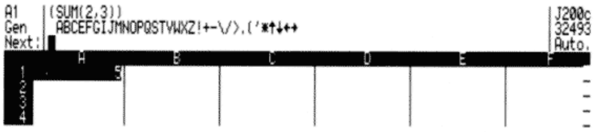
One feature you will find especially useful is that you could have put 2*15 or 4*A1 or any other expression in the brackets part of the function. The part within the brackets is calculated first and then the result is fed to the function. You can even have functions of functions such as SIN(SIN(30)) and so on. There are no real restrictions except the number of levels of brackets which are limited to five in any one expression.

For example:



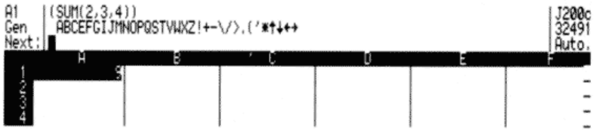
```
A1    |(SIN(SIN(30)))                      |J200c
Gen   | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→  |32489
Next: |                                    |Auto.
          A        B        C        D        E        F
 1  0.008726535
 2
 3
 4
```

The last type of function is the one which has two or more values enclosed in ( ). These enclosed values are called arguments. For example, **SUM(*n1*,*n2*)** adds all the values separated by commas in the brackets. This function returns a value of *n1+n2*. Try these functions for yourself:



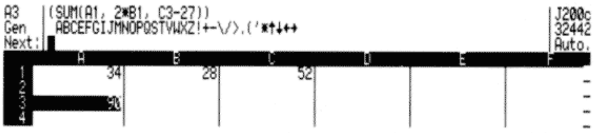
```
A1    |(SUM(2,3))                          |J200c
Gen   | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→  |32493
Next: |                                    |Auto.
          A        B        C        D        E        F
 1             5
 2
 3
 4
```

You can have as many parts inside the brackets as you like. Here is an example with three arguments:



```
A1    |(SUM(2,3,4))                        |J200c
Gen   | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→  |32491
Next: |                                    |Auto.
          A        B        C        D        E        F
 1             9
 2
 3
 4
```

The values could also be expressions:



```
A3    |(SUM(A1, 2*B1, C3-27))              |J200c
Gen   | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→  |32442
Next: |                                    |Auto.
          A        B        C        D        E        F
 1            34       28       52
 2
 3            90
 4
```

You can see the full list of functions in the Expression Entry section of the command reference further on in the manual.

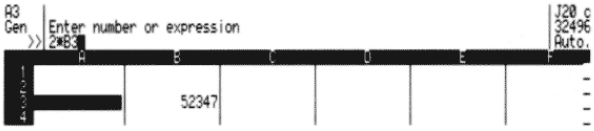
If you want to enter a function such as SUM(A1,A2,A3,A4) to add the cells from A1 to A4 you can use the shorthand SUM(A1...A4). A1...A4 is a range meaning use all the values between the first co-ordinate and the second. You only have to enter the first full stop and the spreadsheet will add the other two for clarity; you just type **A1.A4** (remember upper case!).

You can even use the function in the form SUM(A1...A4,B7). Use a range anywhere you would otherwise put a list of adjacent entries. For most functions the range will still work if one or more of the entries is a blank. This is because the functions only work on the non-blank cells.

These, then, are what make expressions. There are just two more facilities you might find useful. The first is the **#** symbol which, when used after you have typed a co-ordinate, puts its value into the worksheet. If you type **2+A1#**, # will put the value of A1 into the expression rather than the cell co-ordinate 'A1'. If you type # and A1 is 3 you will find your expression becomes 2+3.

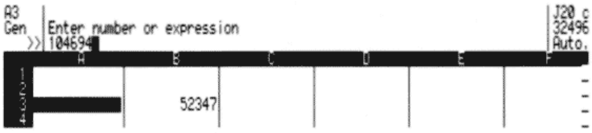
This means that if you later change the value that is held in cell A1 it will make no difference to your new expression.

The other facility is the **!** symbol which forces the expression you are entering to be replaced by its value up to that point. So if you type **2+3!** the display will immediately convert to 5 and you can continue the entry from that point. If you force a value that includes some cell references e.g. 2*B3 this is replaced by a numerical constant and the value will not change even if B3 is altered.



```
A3    |                                    |J20 c
Gen   |Enter number or expression          |32496
   >> |2*B3                                |Auto.
          A        B        C        D        E        F
 1
 2
 3               52347
 4
```

When **!** is pressed instantly becomes:



```
A3    |                                    |J20 c
Gen   |Enter number or expression          |32496
   >> |104694                              |Auto.
          A        B        C        D        E        F
 1
 2
 3               52347
 4
```

Both of these commands have the effect of making the expressions independent of changes in the data held in the cells referred to.

The order of calculation

A powerful feature of the Notebook spreadsheet is that you will find it does not matter where in the worksheet you enter your data or expressions. If you have used other spreadsheets before you will appreciate the difficulties of getting all your calculations in the right order and avoiding forward references.

As you enter an expression the Notebook spreadsheet gives it an 'order of calculation number'. This is not affected by where you enter the expression, just by the references in the formula. If you enter 2*B1 into A3, for example, then you know that you want A3 to be calculated after B1 otherwise it does not make sense. Well, the Notebook spreadsheet goes through all the expressions as you enter them, working out which is the correct order for calculation.

If the entry is a constant then it does not have an 'order of calculation number'.

This feature allows you to rearrange the sheet as you like without affecting your calculated values. The operation is entirely automatic so you don't have to worry about it.

You will be able to see the calculation numbers displayed in angled brackets if you use the ☒ for eXchange command. Those cells which contain a low order of calculation number are calculated before those with a high number.

Recap:

We have looked at some of the possible expressions that can be entered into the Notebook spreadsheet. It is these that make it such a powerful tool.

Each expression is assigned a number that tells the spreadsheet which order they should be calculated in.

Tutorial VII

Mathematical Functions

Here is a more detailed list of the various mathematical functions available. Again, you need not worry about those functions you are unlikely to use.

This tutorial is designed to show you how to use functions in the Notebook spreadsheet and which are available - there will be no attempt to explain the meaning of the mathematical terms available.

Remember, as you read, that arguments are what you put in brackets after the function name and which the function is expected to use to work out its current value.

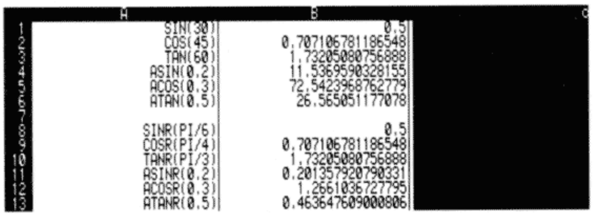
The usual trigonometric functions are available and there are two versions of each for degrees and radians.

The functions **SIN**, **COS**, **TAN**, **ASIN**, **ACOS** and **ATAN** refer to degrees. Those with **A** in front represent the inverse values of those without.

SINR, **COSR**, **TANR**, **ASINR**, **ACOSR** and **ATANR** are the equivalent functions using radians.

Each function takes just one argument enclosed in brackets.

So that you can see what is going on with this screen example, the expressions in column B have been typed into column A in text form, remember that although the two look the same the text column contains only 'words' which cannot be calculated.



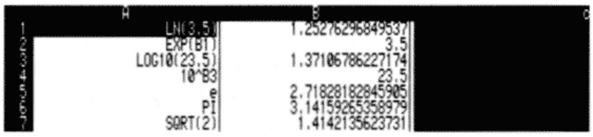
	A	B
1	SIN(30)	0.5
2	COS(45)	0.707106781186548
3	TAN(60)	1.73205080756888
4	ASIN(0.2)	11.5369590328155
5	ACOS(0.3)	72.5423968762779
6	ATAN(0.5)	26.565051177078
7		
8	SINR(PI/6)	0.5
9	COSR(PI/4)	0.707106781186548
10	TANR(PI/3)	1.73205080756888
11	ASINR(0.2)	0.201357920790331
12	ACOSR(0.3)	1.2661036727795
13	ATANR(0.5)	0.463647609000806

The natural logarithm is referred to as **LN** and the base 10 logarithm is **LOG10**. The natural antilogarithm is defined mathematically as e^x (the exponential constant *e* multiplied to the power of *x*) and is referred to as **EXP**.

The equivalent base 10 antilogarithm must be obtained by using 10^*x* where *x* is the value for which you want the antilogarithm.

The exponential constant *e* is available as a function without an argument **e** (lower case is essential when you type this). Similarly **PI** is available.

The square root is called with **SQRT**.



	A	B
1	LN(3.5)	1.25276296849537
2	EXP(B1)	3.5
3	LOG10(23.5)	1.37106786227174
4	10^B3	23.5
5	e	2.71828182845905
6	PI	3.14159265358979
7	SQRT(2)	1.4142135623731

Utility Functions

Sum, minimum and maximum, available as **SUM**, **MIN** and **MAX** will scan a list and return the relevant value. **COUNT** will find the number of non blank entries in a list. All these functions are of the form *FUNC*(list).

SUMIF(*crd1*, *crd2...crd3***)** is a conditional adding up function. Only those items in the range *crd2...crd3* will be included that correspond to a cell entry in the column specified by *crd1* and are on the same line.

	A	B
1		123.45
2	MARKER	23.45
3		245.56
4	MARKER	45.67

SUMIF(A1, B1...B4) gives the answer 69.12
SUM(B1...B4) gives the answer 438.13

It should be noted that *crd1* may point to any cell in the marker column but must not point to a cell with a text entry in it or an error message will result.

BLEEP Sets off a bleep. Use it as an audible signal.

GROW takes two arguments, a value and a percentage. The effect of GROW is to cause a percentage increase to the value. The form is GROW(*value, percent*). Despite the name, the percentage can be negative.

The following functions in this section are of the form *FUNC*(*value*).

The **ABS** function gives the absolute value of an argument, that is, the value ignoring the sign.

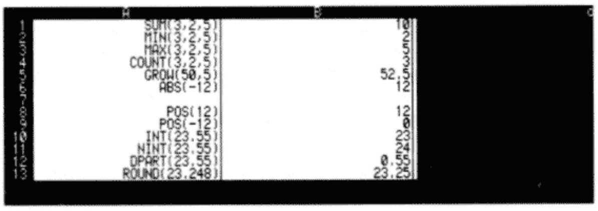
POS returns the value of the argument if it is positive but zero if it is negative. This is useful in tax calculations.

The integer (non-decimal) part of a number can be obtained with **INT** or if you want the nearest whole number to a value use **NINT**. The decimal part of a number is given by the function **DPART**.

ROUND is a useful function, it will round a value to 2 decimal places.

NOTE: it is very different from most of the functions in that it puts this rounded value back into memory and the original value is lost. You may need this function in complex financial calculations which otherwise would not balance because fractions of a penny (cent etc.) are having an effect.

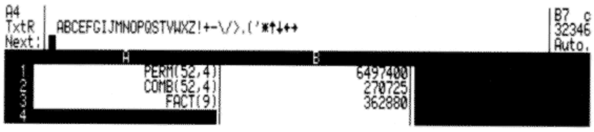
RND returns a true integer random number between 0 and 127. A new value is picked each time the sheet is calculated.



	A	B
1	SUM(3,2,5)	10
2	MIN(3,2,5)	2
3	MAX(3,2,5)	5
4	COUNT(3,2,5)	3
5	GROW(50,5)	52.5
6	ABS(-12)	12
7		
8	POS(12)	12
9	POS(-12)	0
10	INT(23.55)	23
11	NINT(23.55)	24
12	DPART(23.55)	0.55
13	ROUND(23.248)	23.25

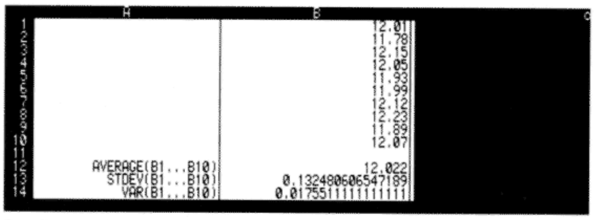
Statistical Functions

Permutation and combination can be calculated with **PERM(***n*, *r***)** and **COMB(***n*, *r***)** where *n* is the total number of items and *r* is the number to be combined or permed. For factorial use **FACT(***value***)**.



```
A4    |                                         |B7  c
TxtR  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓++        |32346
Next: |                                         |Auto.
           A                        B
1          PERM(52,4)          6497400
2          COMB(52,4)           270725
3             FACT(9)           362880
4
```

Standard deviation and variance and average all act on a list to complete your set of powerful statistical tools. These functions are called **STDEV**, **VAR** and **AVERAGE**. (See the magazine BYTE Nov 1983 pp560-563 for the method used).



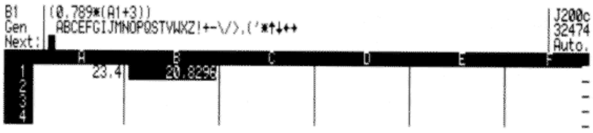
```
                A                        B               c
1                                    12.01
2                                    11.78
3                                    12.15
4                                    12.05
5                                    11.93
6                                    11.99
7                                    12.12
8                                    12.23
9                                    11.89
10                                   12.07
11
12       AVERAGE(B1...B10)          12.022
13         STDEV(B1...B10) 0.132480606547189
14           VAR(B1...B10) 0.0175511111111111
```

Defined Functions

A defined function works just the same way as a built in function does except that it uses a calculation that you have typed in yourself. It is therefore a way in which you can set up a complex formula once that you may want to use many times.

It is a single cell formula that can be used by other cells and, of course, after the first time you do not have to write the formula out in full again. Consider the following formula, 0.789*(A1+3), which uses the value held in cell A1. If you had to write it out 50 or 100 times it would get very time consuming and also quickly use up your available memory.

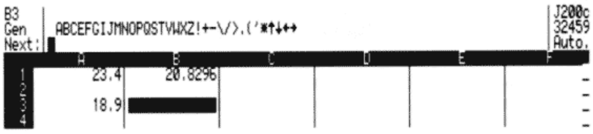
Let us try an example - set up a sheet as follows with a constant in A1, e.g. **23.4**. Put the expression **0.789*(A1+3)** in B1:



```
B1    |(0.789*(A1+3))                           |J200c
Gen   | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓++        |32474
Next: |                                         |Auto.
        A        B        C        D        E        F
1      23.4   20.8296
2                                                    -
3                                                    -
4                                                    -
```

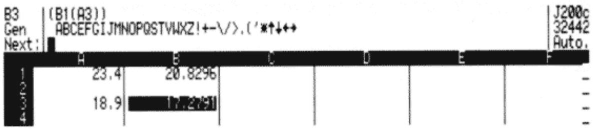
To use a defined function you only have to give the co-ordinate of the cell which contains the formula and follow it with another cell reference so that it can get the value on which you want it to operate. To call the function, write the formula cell reference and follow it by a bracket, just as if you were writing a built in function. Inside the brackets put the co-ordinate of the cell with the value that you want passed to the function, or put a constant.

Put a different constant, e.g. **18.9**, in A3 and move to cell B3. This is the cell where we will call our defined function.



```
B3    |                                         |J200c
Gen   | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓++        |32459
Next: |                                         |Auto.
        A        B        C        D        E        F
1      23.4   20.8296
2                                                    -
3      18.9                                          -
4                                                    -
```

In the cell B3 enter **B1(A3)** - you can read it as: take the formula in B1 and replace the first cell reference found by A3. This is what is done.



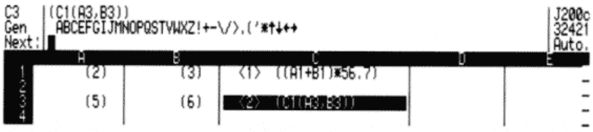
```
B3    |(B1(A3))                                 |J200c
Gen   | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓++        |32442
Next: |                                         |Auto.
        A        B        C        D        E        F
1      23.4   20.8296
2                                                    -
3      18.9   17.2791                                -
4                                                    -
```

In the example the formula has acted on A3 to give the answer 17.2791. Note that A1 and B1 are changed when the sheet is recalculated (press **!** to see this).

If there is more than one cell referred to in the original formula then you must have extra arguments in brackets for the defined function to use. Note that if the same cell is referred to more than once in the formula then it must also be repeated the correct number of times in the argument list.

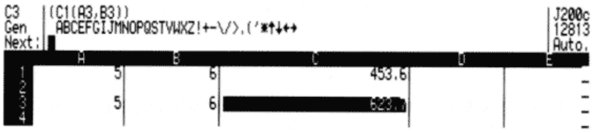
The number of arguments can be less than those in the formula. If this is the case the later cell references will remain unchanged from whatever they were typed in as when you first defined your formula. This can be useful.

It is important that the cell containing the formula for the defined function is kept separate from the rest of the sheet. This is because every time the formula is re-used the cell values referred to by the original formula are changed to those given by you in the argument. If the value of the defined function cell was used in calculations by any other cell in the sheet, undesirable results will occur.



```
C3    |(C1(A3,B3))                              |J200c
Gen   | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓++        |32421
Next: |                                         |Auto.
        A          B          C                  D          E
1      (2)        (3)    <1>  ((A1+B1)*56.7)
2                                                           -
3      (5)        (6)    <2>  (C1(A3,B3))                   -
4                                                           -
```

The above is an example where there are two cell references in C1. As a reminder, to get this type of display just type X ↵ and the formulae will be shown instead of the results. The defined function call will be made in C3:



```
C3    |(C1(A3,B3))                              |J200c
Gen   | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓++        |12813
Next: |                                         |Auto.
        A          B          C                  D          E
1         5          6             453.6
2                                                           -
3         5          6             623.7                    -
4                                                           -
```

Note that the original values in A1 and B1 are changed by the call. You will see this if you press **!** to recalculate the sheet.

Recap:

You have now had an introduction to many of the functions that are available for use in your expressions and should be beginning to get an idea of the power of the Notebook spreadsheet. We have looked in more detail at some of the many functions that are available to you. In particular, mathematical and statistical functions, some general purpose utility functions and user defined functions. Later in this tutorial we will be coming back to look at some of the more advanced types of functions and expressions and what can be done with these.

First, however, there are some more basic commands that have to be dealt with so that you can become completely familiar with the housekeeping of your worksheet and how to manipulate the data.

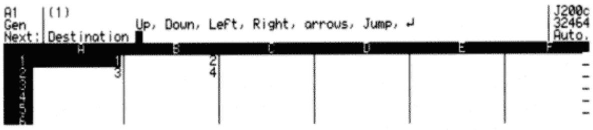
Tutorial VIII - More Techniques

Reorganising the sheet - Copy and Move

The Move command

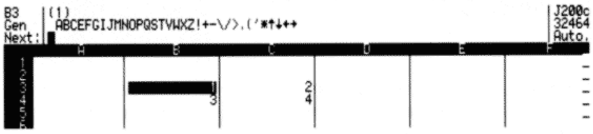
You use the Move command to transfer parts of the sheet to another part of the work area.

Start with a blank worksheet (you should know how to do this by now). Fill up the four top left entries again using the `.' (Entry) command. Now type M B **A1** . **B2** ↵ (Move Block of range A1 to B2):



```
A1    |(1)                                      |J200c
Gen   |           Up, Down, Left, Right, arrows, Jump, ↵  |32464
Next: |Destination                              |Auto.
        A        B        C        D        E        F
1             1        2
2             3        4                             -
3                                                    -
4                                                    -
5                                                    -
6                                                    -
```

You are then asked for the destination of the move operation. Type J **B3** ↵ (Jump to cell B3). The defined block will 'jump' with the cursor:



```
B3    |(1)                                      |J200c
Gen   | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓++        |32464
Next: |                                         |Auto.
        A        B        C        D        E        F
1                                                    -
2                                                    -
3                      1        2                    -
4                      3        4                    -
5                                                    -
6                                                    -
```

Try the Move command with some of the other options, e.g. line and column. The only option not available with the Move command is ALL.

The Move command will overwrite any data in the destination area. Because there is therefore some danger of you mistakenly destroying the sense of any calculations on the sheet, the spreadsheet will make a check and prevent you doing so. However, remember it will let you overwrite some data if it appears that no other parts of the sheet depend on it.

If there are any co-ordinate references in expressions being moved you will find these automatically adjusted in the new locations to take account of the new sheet layout.

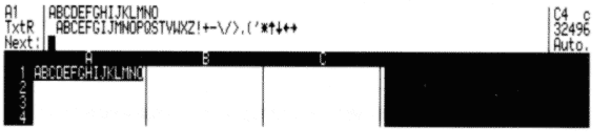
The cells that are moved take their own format definitions with them.

You cannot move or copy cells outside the sheet as it is currently defined. You will just receive the warning message "Cannot multiple move there". You may have to insert some new lines or columns to make room to move the block into.

Copying parts of the sheet

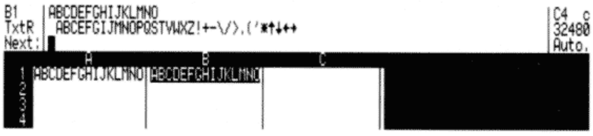
The COPY command can be likened to that known as 'replicate' in other spreadsheet software.

As well as moving data from one part of the sheet to another it will probably come as no surprise that you can copy parts of the data. Start from scratch by Zapping the current sheet. Type [I]3[C]15[↵][T][R][↵] for 'Insert 3 Columns of width 15, Text, Right justified format' then [I]3[L][↵][↵] for 'Insert 3 lines'. Then type [>] and put the entry **ABCDEFGHIJKLMNO** into cell A1.



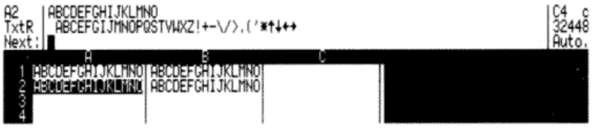
```
A1    |ABCDEFGHIJKLMNO                                   |C4  c
TxtR  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→                 |32496
Next: |                                                  |Auto.
            A               B               C
  1 ABCDEFGHIJKLMNO|               |               |
  2
  3
  4
```

Now let us use the Copy Entry command sequence by typing [C][E][↵]. In this case use the arrows to move the cursor right when asked for the destination, then press [↵]:



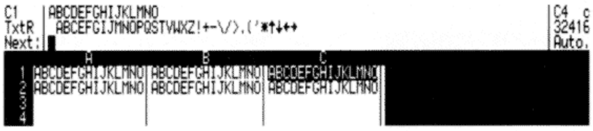
```
B1    |ABCDEFGHIJKLMNO                                   |C4  c
TxtR  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→                 |32480
Next: |                                                  |Auto.
            A               B               C
  1 ABCDEFGHIJKLMNO|ABCDEFGHIJKLMNO|               |
  2
  3
  4
```

Now let us try Copy Line. Type [C][L][↵]. Move the cursor down for the destination then press [↵]:



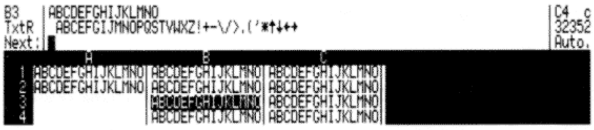
```
A2    |ABCDEFGHIJKLMNO                                   |C4  c
TxtR  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→                 |32448
Next: |                                                  |Auto.
            A               B               C
  1 ABCDEFGHIJKLMNO|ABCDEFGHIJKLMNO|               |
  2 ABCDEFGHIJKLMNO|ABCDEFGHIJKLMNO|               |
  3
  4
```

Now Copy Column. With the cursor in column B press [C][C][↵]. Move the cursor right to column C for the destination and press [↵]:



```
C1    |ABCDEFGHIJKLMNO                                   |C4  c
TxtR  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→                 |32416
Next: |                                                  |Auto.
            A               B               C
  1 ABCDEFGHIJKLMNO|ABCDEFGHIJKLMNO|ABCDEFGHIJKLMNO|
  2 ABCDEFGHIJKLMNO|ABCDEFGHIJKLMNO|ABCDEFGHIJKLMNO|
  3
  4
```

Now try Copy Block defining the block as a rectangle from cells A1...B2. Move the cursor down and across to cell B3:

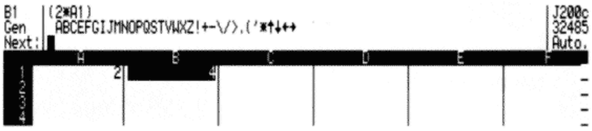


```
B3    |ABCDEFGHIJKLMNO                                   |C4  c
TxtR  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→                 |32352
Next: |                                                  |Auto.
            A               B               C
  1 ABCDEFGHIJKLMNO|ABCDEFGHIJKLMNO|ABCDEFGHIJKLMNO|
  2 ABCDEFGHIJKLMNO|ABCDEFGHIJKLMNO|ABCDEFGHIJKLMNO|
  3                |ABCDEFGHIJKLMNO|ABCDEFGHIJKLMNO|
  4                |ABCDEFGHIJKLMNO|ABCDEFGHIJKLMNO|
```

In some ways the Copy command is the same as the Move command, only the contents of the original location are not deleted. There is, however, one major difference. If you have co-ordinates in your expressions being copied you will be asked whether you want the references adjusted. If you answer "Yes" and it is, for example, a row copy, every reference to other locations on that row will be changed to take account of the new location.

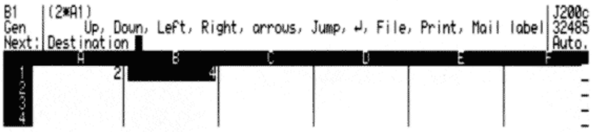
You can try an example of an adjusted copy to get a feel for what is involved. Type [Z][^][↵][Y][↵] to completely clear the sheet and then [M][↵] to make a blank sheet of general format cells.

Type [>]2[↵] to put 2 into location A1, move the cursor to B1 then enter [>]**2*A1**[↵]:



```
B1    |(2*A1)                                              |J200c
Gen   | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→                  |32485
Next: |                                                    |Auto.
         A        B        C        D        E        F
  1      2|       4|        |        |        |        -
  2                                                    -
  3                                                    -
  4                                                    -
```

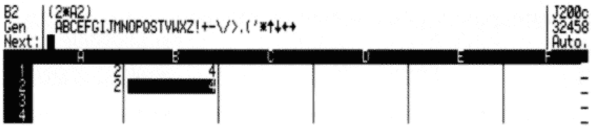
Then type [C][L][↵] for copy line:



```
B1    |(2*A1)                                                        |J200c
Gen   |    Up, Down, Left, Right, arrows, Jump, ↵, File, Print, Mail label|32485
Next: |Destination                                                   |Auto.
         A        B        C        D        E        F
  1      2|       4|        |        |        |        -
  2                                                    -
  3                                                    -
  4                                                    -
```

Followed by [↓] to move the cursor down to the destination line then press [↵].

When you press [↵] you will then be queried about adjusting references; it will not happen automatically, so type [Y][↵]:



```
B2    |(2*A2)                                              |J200c
Gen   | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→                  |32458
Next: |                                                    |Auto.
         A        B        C        D        E        F
  1      2|       4|        |        |        |        -
  2      2|       4|                                   -
  3                                                    -
  4                                                    -
```

You can see that what was **2*A1** in B1 has become altered to **2*A2** in B2. If you think about it, the adjustment option has kept the meaning of the first line, i.e. 'two times the value on the left', in the second line. You will find it is very much the same for columns.

If you had pressed [N] when asked **Adjust references** the formula in B2 would have remained 2*A1.

So far you have seen that Copy works the same as MOVE, except that the original data remains in the position it first was. The copied data overwrites anything that was already in the new position.

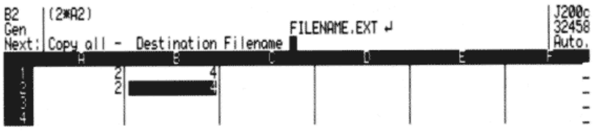
However, you will now see that the Copy command is much more flexible than that - it can be used to copy data to a file or to the printer as well.

The option you would use to do this is Copy All. The Notebook spreadsheet only uses the ALL option to send the data from the sheet to the Printer (normally or as mail labels) or to a file. Copy All from one part of the sheet to another doesn't make sense.

However, you can of course, copy portions of the sheet, entries, lines, columns or blocks, to a file or the printer as well.

We will look at these in detail soon but for now it does no harm to run through the way data is saved to a file. Note the following sequence carefully.

Type [C][A][F] for Copy All to File.



```
B2    |(2*A2)                                              |J200c
Gen   |                           FILENAME.EXT ↵           |32458
Next: |Copy all -  Destination Filename                    |Auto.
         A        B        C        D        E        F
  1      2|       4|        |        |        |        -
  2      2|       4|                                   -
  3                                                    -
  4                                                    -
```

If you type **MYWORK**[↵] when prompted for a file name the whole of the sheet will be copied to a file called MYWORK.MEM.

In a similar way, all the entries could be copied to the printer. The command area and axes markers would be excluded.

Recap:

You have seen how the move and copy commands have many similarities. They can operate on sections of the sheet of variable size to transfer or reproduce data. Both overwrite any data already present at the destination. Both transfer the cell formats together with the cell data.

Tutorial IX

A note on adjusting references - lagged variables

You have seen that when using the Copy command you are prompted with the question:

```
                    Yes, No
Adjust references
```

When using the move command the adjustments are made automatically; you are not asked.

If you are moving or copying a LINE with adjustment then any internal references made to other cells along the same line will be adjusted to reflect the new position.

Similarly, when a column is copied or moved, adjustments are made within the column.

The same is true when a block is copied or moved - all references to cells within the defined block are also changed.

Normally, references to any cells which are outside the moved or copied section will not be adjusted. However, there is a built in option to get round this if you want to. Any reference will be automatically adjusted if you follow the reference to a cell with a single quote (′). The technical term used for this is an offset or lagged variable. In other programs you may see this referred to as a relative cell reference.

For example, a cell reference of A1 in cell B3 will always mean "the cell A1". However, a reference to A1′ in cell B3 means "the cell that is one column to the left and 2 lines above the current cell". If the contents of B3 are copied/moved elsewhere and it contained A1 it would still contain A1. However, if it contained A1′ and was copied or moved to D4, say, the reference would change to C2′ which is the cell that is one to the left and two above D4.

It may not be immediately clear why you would want to do this, or how it works, so just follow this simple example.

Imagine you were preparing a sheet that is designed to carry a

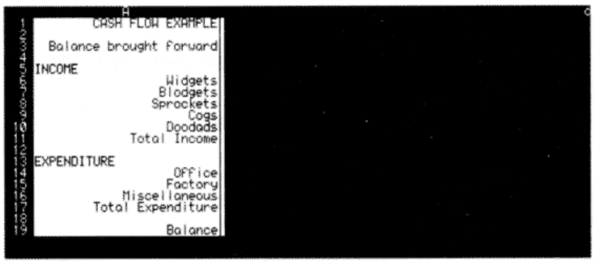
value from the bottom of one column to the top of the next, i.e. each column will contain a cell that refers to a cell in an adjacent column. If any of the columns were copied the reference to the previous column would remain unchanged unless marked by the Quote character, as a result the sense of the sheet would be lost.

The idea will be explained much more clearly if you follow this example for preparing a simple cash flow.

Preparing a simple Cash Flow

One of the most common applications for spreadsheets is budgets and cash flow predictions. You are going to see how to prepare a cash flow sheet using some of the short cuts that the Notebook spreadsheet provides. Even if you aren't interested in financial applications it is worth studying this example because it demonstrates multiple copying, lagged or offset variables and automatic reference adjustment.

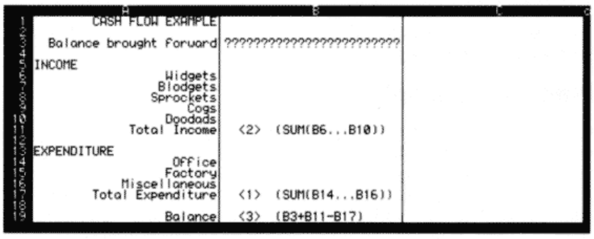
You will normally start by setting up a text column with labels for the various items to be included in your cash flow. Here is a very simplified example. Remember that ⇆ can be used to see more lines in your worksheet and the ↑ and ↓ keys will go up and down a page at a time.



```
              A
 1       CASH FLOW EXAMPLE
 2
 3   Balance brought forward
 4
 5 INCOME
 6                   Widgets
 7                  Blodgets
 8                 Sprockets
 9                      Cogs
10                   Doodads
11              Total Income
12
13 EXPENDITURE
14                    Office
15                   Factory
16             Miscellaneous
17         Total Expenditure
18
19                   Balance
```

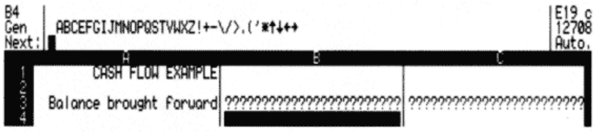
As you can see, there are four basic elements. These are the balance brought forward, the income, the expenditure and the balance for the period. The balance brought forward always refers to the previous period. The other three elements always refer to the current period.

Now you will see the rules that might make up such a sheet. This display was obtained using X for eXchange command. Temporarily, the width of column B has been increased so you can see everything. At this stage, no actual values have been entered.



```
              A                          B                       C
 1       CASH FLOW EXAMPLE
 2
 3   Balance brought forward ????????????????????????
 4
 5 INCOME
 6                   Widgets
 7                  Blodgets
 8                 Sprockets
 9                      Cogs
10                   Doodads
11              Total Income   <2>  (SUM(B6...B10))
12
13 EXPENDITURE
14                    Office
15                   Factory
16             Miscellaneous
17         Total Expenditure   <1>  (SUM(B14...B16))
18
19                   Balance   <3>  (B3+B11-B17)
```

The line of question marks in cell B3 shows that it is used in a following calculation but no value has yet been put in. This is to remind you to do so later. This column is the dummy for the first month. Now you copy these formulae for the second month. With the cursor in column B type C C ↵. Move the cursor right in the answer to Destination then press ↵, answer Y to the **Adjust references** question, and you will arrive at:

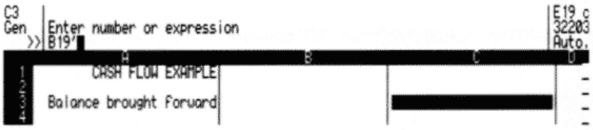


```
B4    |                                                  |E19 c
Gen   | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→                |12708
Next: |                                                  |Auto.
              A                          B                       C
 1       CASH FLOW EXAMPLE
 2
 3   Balance brought forward ???????????????????????? ????????????????????????
 4
```

You can see the balance brought forward for this new column is still undefined. But it is equal to the value in B19 and so you can put B19 in C3. The balance brought forward always relates to the preceding column. You want to set up column C so that you can copy it a few more times for the other months and include the balance brought forward. If you just put B19 in C3 and copy it, then the value B19 will remain unchanged. The Copy command only adjusts the formulae that relate to references up and down the column, other references are unaffected.

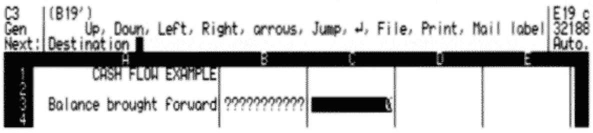
There is a simple way round this problem. Just follow the reference by a single quote mark. This tells the Notebook spreadsheet that

you want the reference to be relative and it should be adjusted whenever a copy is made. So the entry you make in C3 becomes B19′.



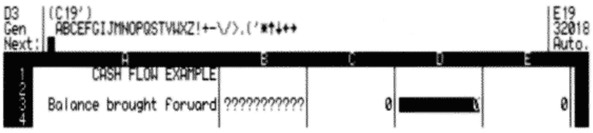
```
C3    |                                                  |E19 c
Gen   |Enter number or expression                        |32203
   >> |B19′                                              |Auto.
              A                          B                C         D
 1       CASH FLOW EXAMPLE                                          -
 2                                                                  -
 3   Balance brought forward                                        -
 4                                                                  -
```

You now have a full dummy column for further copying. Note that the actual number of entries you have made so far is very few. For this example we are just going to set up two further months. This time you specify that you want the column to be copied two times. That is, C C 2↵.



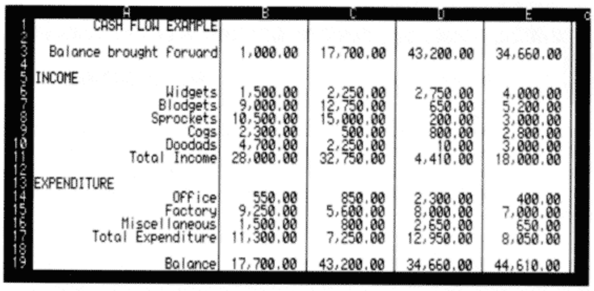
```
C3    |(B19′)                                                        |E19 c
Gen   |    Up, Down, Left, Right, arrows, Jump, ↵, File, Print, Mail label|32188
Next: |Destination                                                   |Auto.
              A                B          C          D          E
 1       CASH FLOW EXAMPLE
 2
 3   Balance brought forward ???????????          0
 4
```

Again you will be asked for a destination and whether you want to adjust references. Make the destination the column to the right and answer Y to the adjust references question.



```
D3    |(C19′)                                            |E19
Gen   | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→                |32018
Next: |                                                  |Auto.
              A                B          C          D          E
 1       CASH FLOW EXAMPLE
 2
 3   Balance brought forward ???????????          0          0          0
 4
```

You can see that the balance brought forward has been adjusted for the new location in the way you would want. With a bit of tidying and the addition of actual figures you will achieve something like this:



	A	B	C	D	E
1	CASH FLOW EXAMPLE				
2					
3	Balance brought forward	1,000.00	17,700.00	43,200.00	34,660.00
4					
5	INCOME				
6	Widgets	1,500.00	2,250.00	2,750.00	4,000.00
7	Blodgets	9,000.00	12,750.00	650.00	5,200.00
8	Sprockets	10,500.00	15,000.00	200.00	3,000.00
9	Cogs	2,300.00	500.00	800.00	2,800.00
10	Doodads	4,700.00	2,250.00	10.00	3,000.00
11	Total Income	28,000.00	32,750.00	4,410.00	18,000.00
12					
13	EXPENDITURE				
14	Office	550.00	850.00	2,300.00	400.00
15	Factory	9,250.00	5,600.00	8,000.00	7,000.00
16	Miscellaneous	1,500.00	800.00	2,650.00	650.00
17	Total Expenditure	11,300.00	7,250.00	12,950.00	8,050.00
18					
19	Balance	17,700.00	43,200.00	34,660.00	44,610.00

Tutorial X

Saving, Loading and Printing - A further look at the Copy command

When you create a new worksheet and type some data into it, the sheet you are working on is held in the spreadsheet program's own working memory. When you want to finish working on the data you must save what you have done into a "file". A file is just a computer word that means a collection of data that has been stored away for future use. In the word processor you are probably already aware that it stores what you type in "documents". Well, documents could also be referred to as "files" but in the case of the word processor the word "document" is more meaningful.

The difference between the word processor and the spreadsheet is that the word processor always asks you for a name before you start and then automatically stores what you have done when you press Stop. In the spreadsheet, things work the other way round. You work on a sheet and then, at the end, you must use the Copy command to copy it to a file and give it a name at that time.

If you just leave the spreadsheet by pressing Stop or using the Q command before you have explicitly saved your work, it is still stored away for you in a file with the special name of RESTART.MEM. Later, when you next use the spreadsheet, you can reload this file very quickly by selecting the Restart option from the opening menu of the spreadsheet. You should not rely on this as the only means of saving as RESTART.MEM is over-written every time you leave the spreadsheet. It is far better to get into the habit of saving your work to a named file, using the Copy command, each time before you leave the spreadsheet.

To copy the data you have entered to a file give the destination as F for file. You will then be asked for a filename. Unlike the word processor, the spreadsheet imposes some limits on the names you can use. This is mainly because it always uses the same characters for the last 4 characters of the name. (Either .MEM, .TXT, .DIF or .DAT). This means that you can only actually give up to 8 characters, and a further restriction is that these may only include any printable character except $*?=/.,: or space. The additional 4 characters are called an extension.

As you will see later, the spreadsheet treats each of these four types of file in slightly different ways and expects the information in them to conform to certain types. It is only by this extension that the spreadsheet knows what sort of data is held in a file so you must always make sure you select the correct one.

Valid names for spreadsheets might be:

SALES.MEM
BUDGET.DAT
INCOME.DIF
DEBTORS.TXT

The following names would not be valid:

JANUARYSALES.MEM	too long, must be 8 characters.
JAN SALES.MEM	no space allowed in name

If you are saving or loading .MEM (normal spreadsheet files) then you don't need to type the .MEM. For any of the other three types of file you must give the relevant extension. By doing this you will inform the spreadsheet of the type of file you want to read or write and, if reading, it will expect the file to contain information of a certain type.

The command sequence to load a file is C F *name* i.e. Copy File *name* (to the sheet).

To save a file the sequence is C A F *name* i.e. Copy All of the sheet to the File *name*.

Of course, you can save sections of the sheet such as a block by specifying alternatives to the All command e.g. C B **A1** . **D3** ↵ F *name*.

If you wish to insert a file into a sheet that already exists it is necessary to create room for the incoming file. Alternatively, the incoming file can be simply appended to the existing one.

Blocks that are saved or loaded (or blanked) must be self contained, the formulae within must make no references to cells that are outside of the block.

It may be necessary to ensure that the defined formats of an existing sheet are of the correct type for the incoming file.

Let us look in more detail at the four types of file.

MEM

If you wish to save all or part of the worksheet in such a way that the Notebook spreadsheet can read it and display it in the original form then use the .MEM (for memory) extension.

You do not normally have to type .MEM as this is added automatically if you do not give an extension. This is the file type that you will use most of the time for day to day saving and loading of the sheets you work on.

Note also that the .MEM file is the only type that can be used to reliably load in a sheet that has not already been defined. It is the only option that will let you load in rows or columns that are empty of data. It can therefore be used for loading a predefined worksheet matrix into which you need only insert the data - a useful example would be in the preparation of yearly accounts that conform to a regular pattern.

If you intend to send or receive spreadsheet files to/from another computer and you use the spreadsheet program called "The Cracker" on that machine then you should use the .MEM file format as all information will then be preserved. The Cracker is the spreadsheet program on which the Notebook spreadsheet is based. Versions are available for Amstrad PCW and IBM PC compatible computers.

DIF

Next there is the .DIF file or Data Interchange Format file. This type of file can transfer text and numbers between different types of spreadsheet and is commonly used by graphics packages as means of collecting the data. The Notebook spreadsheet can read and write these files. You should not concern yourself with the internal organisation of these files. The only time you are likely to use .DIF is when sending files from the Notebook to another computer that uses a different type of spreadsheet/graphics program.

TXT

If you want to write out all or a part of your worksheet to a file that can be used by an editor or word processing program, give it a filename that has a .TXT extension. This is very useful if you want to incorporate the results calculated by the Notebook spreadsheet into a full written report. The word processor in your Notebook can read in .TXT files produced by the spreadsheet.

You can also read a .TXT file into the spreadsheet and this provides a useful way of bringing in tables of information from other programs. The numbers and text coming in will need to be in a proper tabulated form as the information is allocated to cells in the sheet depending on its location along each line.

The .TXT file type is the only other example where it is possible to load in data without having first defined a blank worksheet. If you attempt to do this the Notebook spreadsheet will first read the number of lines in the data and assign each of these to a row in the sheet. However, all of the information across the lines will be assigned to one, possibly very large, column. Do make sure that this column is not allowed to exceed 127 characters, which is the absolute maximum column width possible.

Such a text file can be read and edited in the Notebook's own word processor. So, you could enter data in the word processor and then load it into the spreadsheet or you could read results from a spreadsheet into the Notebook's word processor. Remember that the file name you use must be limited to between 1 and 8 characters followed by the extension .TXT for this to work.

DAT

The .DAT extension is used when you wish to read a file of only numbers into your sheet. The file should be in character form such as you would get from an editor or word processor, or formatted output from a programming language such as BASIC. This could, for example, be produced by the Notebook's own word processor or the BBC BASIC that is built into the Notebook.

The file will be read in much the same way as prepared but be careful that if more than one column is being read then zero readings should be shown by a 0 and not just by blanks. It will do no harm but if there was a blank in the first column then the second column will be read as if it was the first. This form of file does not have to be properly tabulated as the lines are scanned to search for the numbers on them. Each number can be separated by spaces or a comma.

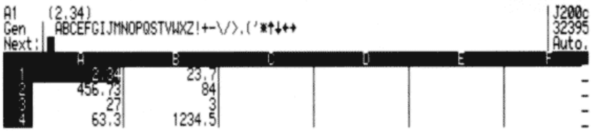
Let us try an example of loading in one of these file types.

Use the Notebook's word processor to create a file called EXAMPLE.DAT which contains this information:

```
2.34, 23.7
456.73, 84
27, 3
63.3, 1234.5
```

Make sure there is a blank line at the end of the list. Then switch to the spreadsheet and create a blank worksheet by typing M ↵. This will definitely be large enough to take the incoming data.

Now use the Copy command to load in the data from the EXAMPLE.DAT file. Type C F **example.dat** ↵. When asked for the destination just type ↵ to indicate you want A1, the current cell, to be the start:



```
A1    (2.34)                                                J200c
Gen  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→                     32395
Next:|                                                      Auto.
          A           B           C           D           E     F
 1      2.34|       23.7|           |           |           |     -
 2    456.73|         84|           |           |           |     -
 3        27|          3|           |           |           |     -
 4      63.3|     1234.5|           |           |           |     -
```

The entries from your incoming .DAT file will be displayed in the default format of the columns. Because the spreadsheet knows that a DAT file contains numbers only, if the default format is text then the General numeric format will be used instead.

Remember that any blank cells must be represented by a zero or the layout may be structured wrongly.

For example if the EXAMPLE.DAT file contains this:

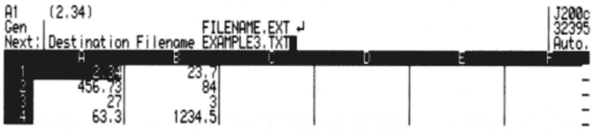
```
            1.23
2.34,       5.67
```

it will be read into the Notebook spreadsheet as

```
====A====        ====B====
     1.23
     2.34             5.67
```

You may write out the data from a worksheet to a .DAT file to be used on another computer. However, such a file cannot be edited using the Notebook's word processor. Use a .TXT file if you want to do this. If a .DAT file is written out, any text entries will be converted to 0.0 to maintain alignment of the data. Such a file is comma delimited, each piece of data is separated from the next by a comma. This is a format that BASIC and other programming languages can easily read (if this means nothing to you don't worry, it is another piece of computing convention).

Type C B A1 . B4 ↵ F **EXAMPLE3.TXT** ↵ to copy the block of numbers to a file called EXAMPLE3 which has a text type.



```
A1    (2.34)                                                J200c
Gen  |                      FILENAME.EXT ↵                  32395
Next:|Destination Filename EXAMPLE3.TXT                     Auto.
          A           B           C           D           E     F
 1      2.34|       23.7|           |           |           |     -
 2    456.73|         84|           |           |           |     -
 3        27|          3|           |           |           |     -
 4      63.3|     1234.5|           |           |           |     -
```

Now switch to the word processor's list of stored documents by pressing Function-L and select the file called EXAMPLE3.TXT. You will see:

```
  2.34       23.7

456.73         84

    27          3

  63.3     1234.5
```

Unlike a .DAT file, commas are not added in a text file. Entries are just spaced out into columns. Also note that it was purely giving a .TXT, rather than a .DAT extension that caused the different file format to be used.

With files stored using the .TXT extension, only the body of the worksheet will be in the file, not the control area, column letters or line numbers.

The .DAT and .TXT file extensions cause very similar files to be produced except that in .DAT files any text is replaced by a 0 to maintain alignment and items are separated by commas. TXT files can be used in text editing programs such as a word processor while .DAT files can be used in programs that just want the numerical values to process them in some way.

When using the Copy All command to save worksheets to either DAT, DIF or TXT files remember that every cell in the defined sheet, even if it is blank, will cause spaces or zero entries to be written to the file. So, if you started with the Make blank sheet command which makes a sheet with 10 columns and 200 lines then, even if you only have a few entries on it, the resulting saved files could be huge. It is best to use the Copy Block command to only copy selected areas when saving data to files of these types.

Printing from the Notebook Spreadsheet

The Copy command is also used for printing data onto paper by specifying P for printer as the destination. Any specified Entry, Line, Column, Block or All of the sheet can be printed. The data that is printed will appear without any column dividers, status lines or other system messages. It will be spaced and displayed in a way that conforms to the existing display format, column widths etc.

In order to fit a lot of data on a page you may wish to make use of the OUT command to send control characters to your printer. Condensed print can then be selected. Consult your printer manual for details of what type faces it is capable of, and which codes are required to select between them.

You will find that a lot of printers (Epson compatible) list character 15 as the sequence to switch into condensed print. Character 18 is used to switch back from condensed printing. You can send this code from the spreadsheet by typing O Control+O ↵. Later, to switch condensed off, use the command O Control+R ↵. You may also find that to switch NLQ printing on and off you must send ESC x 1 or ESC x 0. The "ESC" character is character 27, which you can type as Control-[so, to send an NLQ off sequence you could type O Control+[X 0 ↵.

It is sometimes desirable to be able to print out a copy of the formulae or data as it was entered rather than as it appears under the current format. Use the X for eXchange command to alter the display and then amend the column widths as required. (It is important to remember this last step or the full width of your expressions will not print).

Printing mail labels

If you want to use the Notebook spreadsheet to keep records of names and addresses see also the later sections on Searching and Sorting of data in Tutorial XII which you will probably find useful.

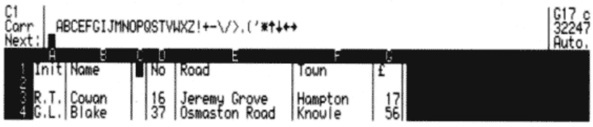
If you do keep lists of names and addresses then you probably will also want to prepare and print address labels. Blank labels are available on listing paper suitable for most printers. The Notebook spreadsheet is able to produce these labels for you. Your address list will, however, need a little preparation in the spreadsheet.

The first stage is to indicate the ends of the lines to be printed. To do this, you must insert columns at the appropriate points, each with a default format of "Carriage Return". A Carriage Return is the code that you send to a printer that signals that you want it to start a new line. This has been included in the possible categories of default formats solely to allow you to control your printer when making labels. You cannot actually enter any data into a column that has been given this format.

There is an example file called ADLIST.MEM in the built-in demonstration files that shows how a list of names and addresses may be stored. The following description is based on that example.

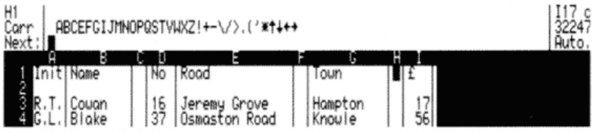
From the opening screen of the spreadsheet type D **ADLIST**↵. This will load in the demonstration list of names and addresses. This is just a typical set of data to show you one possible way in which it could be laid out.

In this example you will not want to print the last column as it only contains an amount of money. You must start by putting a carriage return column in front of column C: Move the cursor to anywhere in column C and type I C **1**↵↵C ↵. This inserts a column of width 1 which has a format type of Carriage Return.



```
C1   |                                        |G17 c
Carr | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→       |32247
Next:|                                        |Auto.
      A     B      C D   E             F        G
   1 Init|Name      | |No |Road         |Town     |£ |
   2
   3 R.T.|Cowan     | |16 |Jeremy Grove |Hampton  |17|
   4 G.L.|Blake     | |37 |Osmaston Road|Knowle   |56|
```

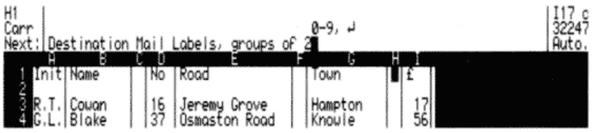
Now two further such columns are needed, one in front of the "town" column (F) and one at the end in front of the "£" column (G):



```
H1   |                                        |I17 c
Carr | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→       |32247
Next:|                                        |Auto.
      A     B      C D   E              F G         H I
   1 Init|Name      | |No |Road          | |Town     | |£ |
   2
   3 R.T.|Cowan     | |16 |Jeremy Grove  | |Hampton  | |17|
   4 G.L.|Blake     | |37 |Osmaston Road | |Knowle   | |56|
```

Now you are in a position to print out your first trial set of labels. You only want to print a portion of the sheet and so you use the Copy block option. Note that the Mail labels option intercepts the carriage return columns that you have set and works out the way the printout should be organised to produce the desired effect - you don't actually get a new line started wherever there is one.

Note that it is absolutely essential for you to make sure that the end of the range is on a Carriage Return Format column. You can see that cell H7 above is such. Omitting this requirement will do no harm but will not give sensible results. You now need to copy a block of the sheet to the printer. Type C B **A3**. **H7**↵. When asked for the destination type M **2**↵ (for Mail labels, 2 across). The names and addresses should now be printed two abreast:



```
H1   |                                        |I17 c
Carr |                        0-9, ↵          |32247
Next:|Destination Mail Labels, groups of 2    |Auto.
      A     B      C D   E              F G         H I
   1 Init|Name      | |No |Road          | |Town     | |£ |
   2
   3 R.T.|Cowan     | |16 |Jeremy Grove  | |Hampton  | |17|
   4 G.L.|Blake     | |37 |Osmaston Road | |Knowle   | |56|
```

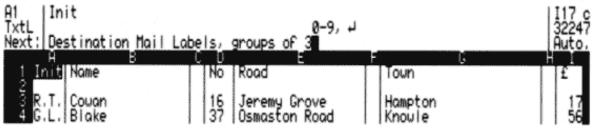
Here is what they will look like. Not very organised, but your next task is to change the width of the columns so that various parts line up under one another:

```
G.L. Blake    A.J. Hunt
37 Osmaston Road    25 Latimer Road
Knowle      Wythall

H.A. Fisher    C.D. Beard
32 Florence Road    86 Valley Road
Henley      Redditch
```

```
R.T. Cowan
16 Jeremy Grove
Hampton
```

After some width adjustment, here is what you can achieve. In this case the labels are going to be printed three abreast:



```
A1   |Init                                    |I17 c
TxtL |                        0-9, ↵          |32247
Next:|Destination Mail Labels, groups of 3    |Auto.
      A     B            C D   E               F G               H I
   1 Init|Name            | |No |Road           | |Town            | |£
   2
   3 R.T.|Cowan           | |16 |Jeremy Grove   | |Hampton         | |17
   4 G.L.|Blake           | |37 |Osmaston Road  | |Knowle          | |56
```

```
G.L. Blake        A.J. Hunt         H.A. Fisher
37 Osmaston Road  25 Latimer Road  32 Florence Road
Knowle            Wythall           Henley

C.D. Beard        R.T. Cowan
86 Valley Road    16 Jeremy Grove
Redditch          Hampton
```

The column widths used in this case are A=4, B=16, C=1, D=3, E=17, F=1 and G=21. The NW command is used to change widths of existing columns.

Further adjustment may be needed with extra lines at the end or changes in spacing to suit the particular labels. You will find that this is quick and easy by trial and error.

Note that if you had used the simple Print option rather than Mail labels the Carriage Return columns would have produced unpredictable and probably undesirable results.

Recap:

We have seen how the Copy command is also used for saving and loading data to/from files in memory and also for printing the data.

When saving or loading, the choice of name extension signals to the Notebook spreadsheet which type of data it should save/load.

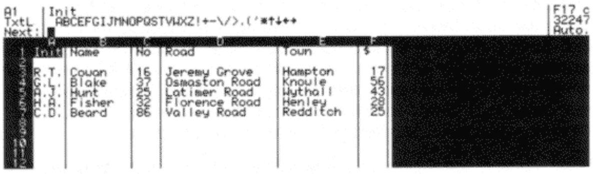
When printing you may have to make use of the OUT command and a special Carriage Return format in order to get the desired effect.

Tutorial XI

Partitioning the screen

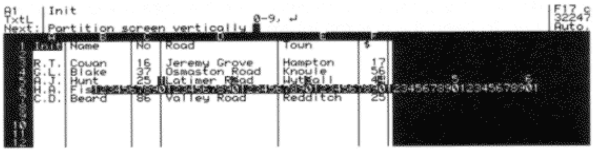
As your worksheet gets larger you will at times want to work somewhere like the bottom right hand corner but still see your line and column headings which are at the top of the sheet. You can do this with the partition commands. These allow you to split the screen horizontally or vertically or both so that you get two, or four smaller displays. The usual term for such split screens is 'windows'.

As well as moving them independently you also have the option of locking the windows so that as you move in one part the other will move in a synchronised way. For example if you scroll the lower window the appropriate headings will scroll past in the top of the screen. It is easy to move around and between the windows with single key commands. One of the demonstration files can be used to experiment with this feature. Clear your previous work with Z A ↵Y ↵ then re-load the example file with D **ADLIST**↵.

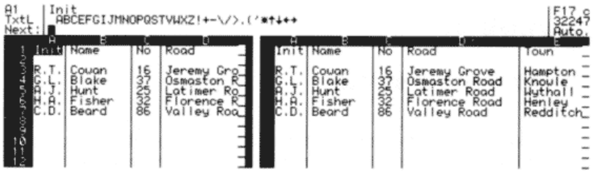


```
A1   |Init                                    |F17 c
TxtL | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→       |32247
Next:|                                        |Auto.
      A     B       C   D            E         F
   1 Init|Name     |No |Road         |Town     |$
   2
   3 R.T.|Cowan    |16 |Jeremy Grove |Hampton  |17
   4 G.L.|Blake    |37 |Osmaston Road|Knowle   |56
   5 A.J.|Hunt     |25 |Latimer Road |Wythall  |43
   6 H.A.|Fisher   |32 |Florence Road|Henley   |28
   7 C.D.|Beard    |86 |Valley Road  |Redditch |25
   8
   9
  10
  11
  12
```

First you are going to see the screen partitioned vertically. As you type P V for Partition Vertically you will see that a line of numbers is put up on the screen. You must use this to judge where you want the split to take place. The number you choose will be the first location of the second window on the screen. You can see that the number 26 is about half way across the screen. Type 26↵ and the second window, to the right, will start at the indicated column 26.



```
A1   |Init                                    |F17 c
TxtL |                        0-9, ↵          |32247
Next:|Partition screen vertically             |Auto.
```



```
A1    |Init                                                  |F17 c
TxtL  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→                     |32247
Next: |                                                      |Auto.
     A     B        C    D                  A     B        C    D               E
  1 Init|Name     |No |Road               Init|Name     |No |Road            |Town
  2
  3 R.T.|Cowan    |16 |Jeremy Gro         R.T.|Cowan    |16 |Jeremy Grove    |Hampton
  4 G.L.|Blake    |37 |Osmaston R         G.L.|Blake    |37 |Osmaston Road   |Knowle
  5 A.J.|Hunt     |25 |Latimer Ro         A.J.|Hunt     |25 |Latimer Road    |Wythall
  6 H.A.|Fisher   |32 |Florence R         H.A.|Fisher   |32 |Florence Road   |Henley
  7 C.D.|Beard    |86 |Valley Roa         C.D.|Beard    |86 |Valley Road     |Redditch
  8
  9
 10
 11
 12
```

You should now have two sets of columns A..D. Don't be confused by seeing the same information in both parts of the screen, it merely means that both windows are set to the same part of the worksheet. You can leave the left view where it is and on a big worksheet move the right view far across the sheet.

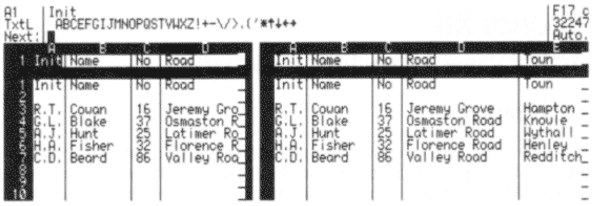
The next command to investigate is the slash [/]. This is used to jump between horizontal window sections. The first time you press it the active cell cursor switches from being to the left of the partition line to the right. Press [/] again and you will see the cursor jumps back to the left window of the display.

The effect as you can see is to jump to the same cell but in the other window. Move the cursor in the right window so that it scrolls left. Notice that the right window continues to display the original position. Press [/] and the cursor jumps back to cell A1 in the left window. If you want both left and right windows to scroll together you must lock them together. To lock them use the [P][S][V][↵] (Partition Synchronised Vertical) command sequence:

Now, as you scroll one of the windows left and right you will see that the contents of the other window follows it.

If you push [⇄] to switch between the normal and enlarged display any horizontal partition will be lost but vertical partitions are maintained.

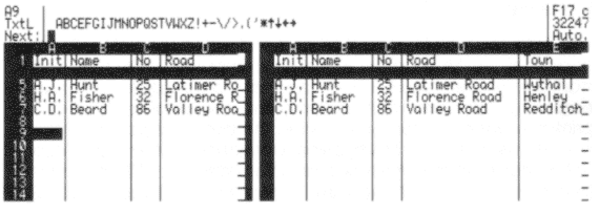
You can divide the screen vertically in a very similar way using [P][H] (Partition screen Horizontally) and specifying the grid number, which is printed in column A of the right hand window. Type **1**[↵] to keep the column labels at the top of the screen:



```
A1    |Init                                                  |F17 c
TxtL  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→                     |32247
Next: |                                                      |Auto.
     A     B        C    D                  A     B        C    D               E
  1 Init|Name     |No |Road               Init|Name     |No |Road            |Town

  1 Init|Name     |No |Road               Init|Name     |No |Road            |Town
  2
  3 R.T.|Cowan    |16 |Jeremy Gro         R.T.|Cowan    |16 |Jeremy Grove    |Hampton
  4 G.L.|Blake    |37 |Osmaston R         G.L.|Blake    |37 |Osmaston Road   |Knowle
  5 A.J.|Hunt     |25 |Latimer Ro         A.J.|Hunt     |25 |Latimer Road    |Wythall
  6 H.A.|Fisher   |32 |Florence R         H.A.|Fisher   |32 |Florence Road   |Henley
  7 C.D.|Beard    |86 |Valley Roa         C.D.|Beard    |86 |Valley Road     |Redditch
  8
  9
 10
```

There are now four parts to the screen. To move between the horizontal parts you use the backward slash [\]:



```
A9    |                                                      |F17 c
TxtL  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→                     |32247
Next: |                                                      |Auto.
     A     B        C    D                  A     B        C    D               E
  1 Init|Name     |No |Road               Init|Name     |No |Road            |Town

  5 A.J.|Hunt     |25 |Latimer Ro         A.J.|Hunt     |25 |Latimer Road    |Wythall
  6 H.A.|Fisher   |32 |Florence R         H.A.|Fisher   |32 |Florence Road   |Henley
  7 C.D.|Beard    |86 |Valley Roa         C.D.|Beard    |86 |Valley Road     |Redditch
  8
  9
 10
 11
 12
 13
 14
```

You can synchronise the parts horizontally or stop it at any time. To get rid of the partitioning just type [P][E] for Partition End. If you save the worksheet to a file it will be in the unpartitioned form when next loaded.

Recap:

After entering headings or explanatory comments into one part of a sheet it is often desirable to be able to see these even when the cursor has moved to a distant part of the sheet.

The Partition command lets you define up to four separate windows on the sheet which can be locked to scroll together or left independent.

Tutorial XII - Database Handling Techniques

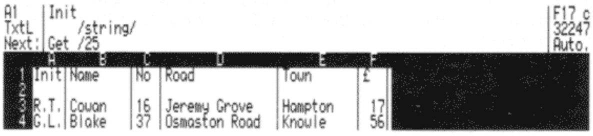
Searching

You may want to find a particular location within a large worksheet without having to go through it looking yourself. To help you, there is the GET command which goes through the columns and lines starting at the current cursor location looking for any piece of data that you care to enter. Note that it does not matter what format the target data is displayed under, only the data as it was entered is searched.

The data you look for must start and end with a special "delineator" character. A "delineator" is just a computer word meaning a start/end marker. Valid delineators that can be used to mark the beginning and end of the string are any characters that are printable but not letters or numerals. The data that you want found will be assumed to have been completed when the second delineator to match the first character after the GET command is found. For example, if you want to find the word SALES you could type [G]**"SALES**. As soon as you type the second " the search is performed. If you were looking for a piece of text that contained " characters you could choose some other delineator. For example, [G]**!"Blodget" Sales!**

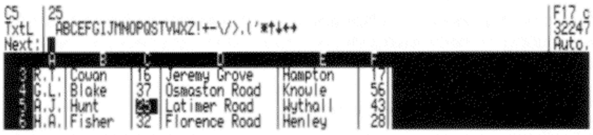
If you want to find a second occurrence of the same string then you only need to type [G] followed by the delineator twice and the string you last used will be automatically inserted between the characters. In the above examples you would just need to type [G]**""** or [G]**!!** to look for a second occurrence.

Load the demonstration file called ADLIST again.



```
A1    |Init                                                  |F17 c
TxtL  |    /string/                                          |32247
Next: |Get /25                                               |Auto.
     A     B        C    D                E           F
  1 Init|Name     |No |Road            |Town       |£
  2
  3 R.T.|Cowan    |16 |Jeremy Grove    |Hampton    |  17
  4 G.L.|Blake    |37 |Osmaston Road   |Knowle     |  56
```

In this example you are searching for the number 25 which is found first at location C5. The delineator used in this case is the / symbol.



```
C5    |25                                                    |F17 c
TxtL  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→                     |32247
Next: |                                                      |Auto.
     A     B        C    D                E           F
  3 R.T.|Cowan    |16 |Jeremy Grove    |Hampton    |  17
  4 G.L.|Blake    |37 |Osmaston Road   |Knowle     |  56
  5 A.J.|Hunt     |25 |Latimer Road    |Wythall    |  43
  6 H.A.|Fisher   |32 |Florence Road   |Henley     |  28
```

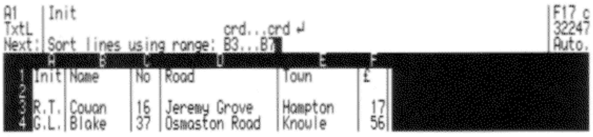
The cursor ends up at the location of the string. Using the command [G]// will find the next occurrence at cell F7.

You should remember that only the actual cell formulae, numeric data and text entries are searched. If, for example, the last column had been in financial format and you had tried to search for 25.00 you would not find it even though it was displayed. As you can see from the contents line only 25 is actually stored in the memory. In practice this means you can only search for things that can be displayed on the contents line.

The Get command is useful for making long jumps across a complicated sheet. You can insert special text entries as markers which help you to quickly find the right place even if insertions and deletions have been made.

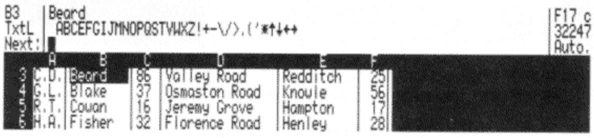
Sorting the lines

The Notebook spreadsheet can selectively sort lines. You can specify the part or whole of a column you want to be used as the basis of the sort. Both text and values can be sorted, either increasing or decreasing. With this facility you can handle address lists and client lists. By using only part of columns in the sorts you carry out many of the activities that you would otherwise use a database management program for. This example assumes you still have the ADLIST example file loaded. In this example the lines are going to be sorted using the name as the basis. There are only valid name entries in the cells B3..B7 so this is specified as the range to be sorted. Type [S]**B3**[.]**B7**:



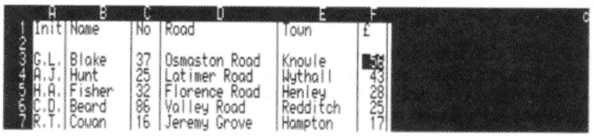
```
A1    |Init                                                  |F17 c
TxtL  |                               crd...crd ↵            |32247
Next: |Sort lines using range: B3...B7                       |Auto.
     A     B        C    D                E           F
  1 Init|Name     |No |Road            |Town       |£
  2
  3 R.T.|Cowan    |16 |Jeremy Grove    |Hampton    |  17
  4 G.L.|Blake    |37 |Osmaston Road   |Knowle     |  56
```

You are now asked whether you want the entries to be sorted into Increasing or decreasing order. Type ' ↵ to specify increasing order.



```
B3    |Beard                                             |F17 c
TxtL  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→                |32247
Next:|█                                                  |Auto.
      A      B        C    D               E          F
 3 C.D.|Beard     |86 |Valley Road    |Redditch  |25|
 4 G.L.|Blake     |37 |Osmaston Road  |Knowle    |56|
 5 R.T.|Cowan     |16 |Jeremy Grove   |Hampton   |17|
 6 H.A.|Fisher    |32 |Florence Road  |Henley    |28|
```

Column B is now in alphabetical order. Next you will see a numerical sort in descending order. Note that the sort is carried out on the internal value (as displayed on the contents line) of the number and not on the numerals as displayed. Type S F3 . F7 ↵ D ↵



```
      A      B        C    D               E          F
 1 Init|Name      |No |Road           |Town      |£ |
 2
 3 G.L.|Blake     |37 |Osmaston Road  |Knowle    |56|
 4 A.J.|Hunt      |25 |Latimer Road   |Wythall   |43|
 5 H.A.|Fisher    |32 |Florence Road  |Henley    |28|
 6 C.D.|Beard     |86 |Valley Road    |Redditch  |25|
 7 R.T.|Cowan     |16 |Jeremy Grove   |Hampton   |17|
```

The lines have now been sorted to make the numbers in column F descending.

Remember that if you do want to keep address lists you will probably also like to be able to prepare mail labels. This was explained in an earlier tutorial.

Recap:

There are two functions, a search command and a sort command, that allow you to reproduce some simple database handling effects using the Notebook spreadsheet.

Tutorial XIII - More complex use of the sheet

More advanced expressions and functions

Here we are going to look at more advanced Functions available to you for use in your expressions. If you feel that we have already covered all of the built in functions that you can take in during the early stages please feel free to skip these sections and concentrate on getting practice with the spreadsheet. However, we do recommend that you try to find time to read about all of the functions later when you are more confident. Remember that the Notebook spreadsheet is designed to be a time saving tool for you, and it may be capable of much more than you had in mind when you first started to use it.

The IF, THEN, ELSE functions

This is a special function group which is known in computing terms as a conditional branch. All this really means is that the outcome of the function changes depending on the result of a test that is made on some portion of the data. It is a method of introducing decision making into the spreadsheet such that the calculations performed vary in their result if certain conditions are or are not met.

This conditional function is of the form:

IF(*expression*), THEN(*expression*), ELSE(*expression*)

The first expression, following the IF, must be logical. 'Logical' is another piece of jargon that just means it must have an answer of TRUE or FALSE. An example of a logical expression is IF (B3=4) which has a value of TRUE if B3 does equal 4 or FALSE if it does not.

If the IF(*expression*) is indeed TRUE the THEN(*expression*) part of the function becomes operative and the current cell takes the value calculated by the expression after the THEN. When the IF part is worked out to be FALSE then the ELSE(*expression*) is used.

The full list of special operators you can use to give you an answer of TRUE or FALSE are:

=	equal
!	not equal (this is the symbol on ^ !)
>	greater than
]	greater than or equal
<	less than
[	less than or equal

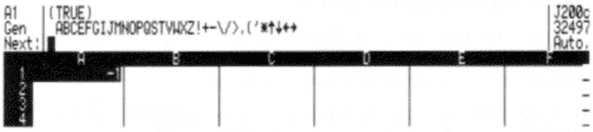
You can also use the functions TRUE or FALSE themselves instead of an expression. In other words you can just type the words TRUE or FALSE in as a cell entry. They do not have arguments so nothing else has to be typed in. Although they are words they count as numeric functions, not as text.

If an expression entry is TRUE it is given a value of -1 and if it is FALSE it is set to 0. Don't worry about why this should be so, the reason lies deep in the roots of the history of computing and in the way that the logical functions work. Just be aware that if you put a logical expression into a cell these will be the displayed values.

Alternatively you can set another cell, say B3, to TRUE or FALSE and then use the conditional in the form IF(B3), THEN(*expr1*), ELSE(*expr2*). It should follow from the above explanation that what this means is IF B3 reads TRUE then calculate expression *expr1*, else (if B3 reads FALSE) calculate expression *expr2*.

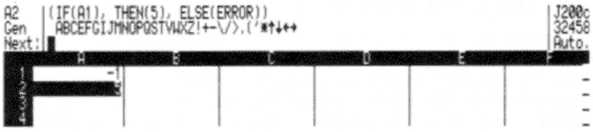
In place of the normal expressions after THEN and ELSE you can use the special function ERROR. If this is encountered during a calculation then the calculation is stopped and a message is put up on the prompt line. You can treat this as if it is a normal error message. No harm can be done using this function. It is a useful method for checking for genuine errors, for bringing macros to an end, or for warning the user that a certain unwanted result has occurred, for example if profits drop below a certain figure.

You can now try an example which includes some of these functions and features. Clear the worksheet and use M ↵ to make a blank one. Now type . **TRUE** ↵ into A1:



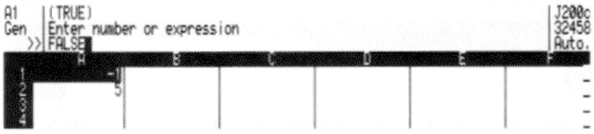
```
A1    |(TRUE)                                            |J200c
Gen   | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→                |32497
Next:|█                                                  |Auto.
        A         B         C         D         E         F
 1           -1|
 2
 3
 4
```

Note that A1 takes on the value -1. Type ↓ to move to A2 and type . **IF (A1), THEN (5), ELSE (ERROR)** ↵ which means if A1 is TRUE then give A2 the value 5 otherwise indicate an error:

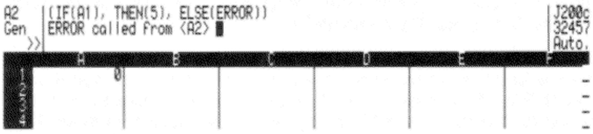


```
A2    |(IF(A1), THEN(5), ELSE(ERROR))                    |J200c
Gen   | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→                |32458
Next:|█                                                  |Auto.
        A         B         C         D         E         F
 1           -1|
 2            5|
 3
 4
```

As A1 was TRUE then A2 has become 5. You can now change A1 to see the effect on A2. Type ↑ . **FALSE** ↵:



```
A1    |(TRUE)                                            |J200c
Gen   |Enter number or expression                        |32458
    >>|FALSE█                                            |Auto.
        A         B         C         D         E         F
 1           -1|
 2            5|
 3
 4
```



```
A2    |(IF(A1), THEN(5), ELSE(ERROR))                    |J200c
Gen   |ERROR called from <A2> █                          |32457
    >>|                                                  |Auto.
        A         B         C         D         E         F
 1            0|
 2
 3
 4
```

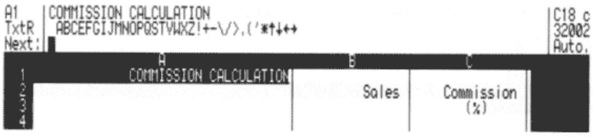
A1 has taken the value 0 for FALSE. Because of the automatic calculation feature, the error message has already come up saying where it was found. In order to make sure that you have noticed it the error message will not let you continue until you have pressed a key. You will also find the current cell has been automatically changed to the one with the ERROR function in it so that you can do something about it.

Table handling functions

Several functions are available to let you extract values from a specified list of cells. They are used in the same way that we would look up and read values from a table or list.

As a first example you are going to see the **LOOKUP** function. This function, when given a value looks along a list to see where this specified value lies and then takes a reading from the adjacent row or column. Consider it as being the same as looking up a value in a printed table where you look for your value in the first column to get your answer in the second, such as logarithm tables.

A typical example of the use of this function may be finding a commission percentage given sales income. These rates tend to jump from band to band. An example of this can be found in the demonstration files called SALES.MEM. Zap any sheet you have loaded and then load the example using the command ⎕ **SALES**↵.

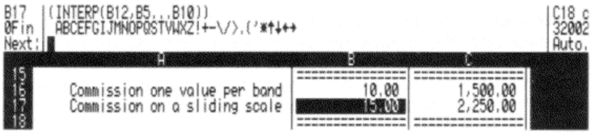


```
A1   |COMMISSION CALCULATION                              |C18 c
TxtR |  ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→                 |32002
Next:|                                                    |Auto.
                   A                        B            C
1            COMMISSION CALCULATION
2                                     |     Sales |  Commission
3                                     |           |     (%)
4                                     |           |
```

In this case the salesperson brought in £15,000 worth of business and so he managed to get into the band between £10,000 and £20,000 for which he gets 10% commission.

The form of the function is **LOOKUP(***value, list***)**, the result returned by the function is taken from the adjacent *list*. You can see it used in cell B16 of the sheet. Type ⎕ **B16**↵ to move directly to it. The value to lookup is in cell B12 and the list of values is in cells B5...B10. The corresponding percentage is taken from the column to the right of the lookup list - that is, column C.

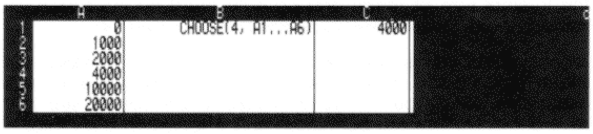
There is a similar function which you can use in the same way called **INTERP** which will interpolate a value from a list. It differs from LOOKUP in that the function tries to work out (interpret) the desired answer even if it is not present in the list. This is used in B17.



```
B17  |(INTERP(B12,B5...B10))                              |C18 c
0Fin |  ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→                 |32002
Next:|                                                    |Auto.
                   A                        B             C
15                                 |===============|===============|
16      Commission one value per band|          10.00|       1,500.00|
17      Commission on a sliding scale|          15.00|       2,250.00|
18                                 |===============|===============|
```

Here the salesperson has been told that the commission will be calculated on a sliding scale based on the sales and commission table. As £15,000 worth was sold this is midway between £10,000 and £20,000 and so he can expect a commission midway between 10% and 20%. The **INTERP** function does this calculation for you and in this case comes up with the answer 15%.

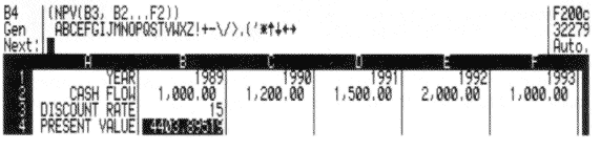
The **CHOOSE** function will look at a list and return the value of the cell in the position in the list given by the first argument. The form of this function is **CHOOSE(***value, list***)**. The value will be rounded to the nearest whole number if it isn't one already.



```
        A                 B                    C
1       0|     CHOOSE(4, A1...A6)|         4000|
2    1000|
3    2000|
4    4000|
5   10000|
6   20000|
```

In this example the CHOOSE function at A8 has looked through A1...A6 to find the 4th cell and returned the value held in it, in this case 4000.

NPV stands for Net Present Value and is a discounted cash flow function that calculates the effect of a discount rate on a set of cash flow figures. The form of the function is **NPV(***rate, list***)** where *rate* is the discount rate in percent and the *list* is a list of cells that contain cash flows. If you do not intend to use the Notebook spreadsheet for financial calculations then there is no need to bother following the next example.



```
B4   |(NPV(B3, B2...F2))                                   |F200c
Gen  |  ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→                  |32279
Next:|                                                     |Auto.
           A          B           C           D           E           F
1          YEAR|       1989|       1990|       1991|       1992|       1993|
2     CASH FLOW|   1,000.00|   1,200.00|   1,500.00|   2,000.00|   1,000.00|
3 DISCOUNT RATE|         15|           |           |           |           |
4 PRESENT VALUE| 4403.89515|           |           |           |           |
```

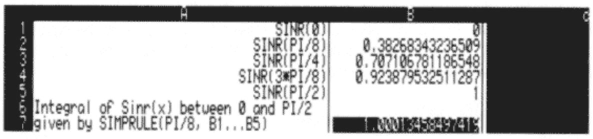
In this example you can assume that in 1993 some money is to be invested and the figures above represent the expected yearly returns on that investment. To find out how the investment will perform, the yearly values each need to be converted to 'present values' and summed. In this instance they are all converted to 1993 values.

The first return in 1994 will be calculated as 1000/(1+dr/100). The 1000 is effectively worth less because of the one year taken to get it. The next year 1200 is obtained but this is worth less still because it is discounted once for 1994 and again in 1995 so its present value is calculated as 1200/(1+dr/100)/(1+dr/100) and so on. The value of the return in *n* years is: return/((1+dr/100)^*n*).

The 'Internal rate of return' is the discount rate necessary to make the present value equal to the initial investment. It can be found by trial and error, changing the value of discount rate until you get the right answer. An example of how they can be automated and the Internal rate of return can be calculated by the Notebook spreadsheet is given in the demonstration file called IRR.

Zap any sheet you are working on and type ⎕ **IRR**↵ to load this sheet. Type ⎕ **A16**↵ to move to the cells where the result is calculated. Then type **!** to force a calculation. You will see that several calculations are performed until it settles on a final value when the trial present value gets close to the target of £5,000. Use of DO, WHILE to perform repeated recalculations is explained in a later tutorial.

If you want to do a numerical integration then you would probably use Simpson's rule. You can use the function **SIMPRULE** to do this directly.



```
                         A                          B
1                               SINR(0)|                 0|
2                            SINR(PI/8)|  0.38268343236509|
3                            SINR(PI/4)|  0.70710678118654B|
4                          SINR(3*PI/8)|  0.923879532511287|
5                            SINR(PI/2)|                 1|
6 Integral of Sinr(x) between 0 and PI/2|                  |
7 given by SIMPRULE(PI/8, B1...B5)      |   1.0001345849741|
```

In this example 5 values of SINR(*x*) have been calculated at intervals of PI/8. The SIMPRULE function has been used to obtain an approximate value of the integral. The exact value is 1. The form of the function is **SIMPRULE(***step, range***)**, the range must have an odd number of values.

A Note on lists

In most functions a list can be specified using a range such as B1...B5. You can, however, have blank entries in your range and the function will still be worked out correctly. This feature allows you to set up a template worksheet and enter your particular data later. It will also cater for the situation where the number of items will be variable.

Date and Time functions

The **DATE** and **TIME** functions may be used in financial calculations, such as tax returns or yearly balance sheets, it is often the case that some account of the time of year has to be taken when determining the required output. By entering values for the time of year automatic adjustments can be made within the program.

Dates in these functions are expressed in the form *ddmm.yyyy*. The dd - day part - can be 1 or 2 digits, the mm - month part - should always be given as two digits so, for example, April is 04. The year should always be expressed as 4 digits. Some valid dates are:

1109.1957	11th September 1957
904.1963	9th April 1963
2407.1993	24th July 1993
2512.1993	25th December 1993

DATEAFTER(*date, days***)**

Gives the date that will be the number of days specified after the input date. So, for example DATEAFTER(312.1992, 120) means - give the date of the day which is 120 days after the 3rd December 1992. The answer is 204.1993, the 2nd April 1993.

DAYSAPART(*date1, date2***)**

Gives the number of days between any two specified dates. For example, if you were born on 9th April 1963 and today's date is 5th August 1992 you could use DAYSAPART(904.1963, 508.1992) to see that you were 10,711 days old (or 257,064 hours or 15,423,840 minutes or 925,430,400 seconds old!)

DAYOFWK(*date*)

Returns the day of week as a number. Saturday has a value 0, Sunday 1, Monday 2 etc. DAYOFWK(508.1992) is 4 which means 5th August 1992 falls on a Wednesday.

DAYOFYR(*date*)

Returns the number of days between January 1st and the present day. For example, DAYOFYR(508.1992) gives the result that 5th August is the 218th day of the year. By dividing this by 7 you can see that it is in week 32.

Note: For the following functions to operate correctly you must make sure the date and time in your Notebook are set before starting the spreadsheet. Setting the date and time was described earlier in the manual.

YEAR

Gives the current year as a four digit number.

MONTH

The current month as a number between 1 and 12

DAY

The day of the month as a number between 1 and 31.

HOUR

The hour in 24 hour format as a number between 0 and 23.

MINUTE

The current minute as a number between 0 and 59.

SECOND

The current second as a number between 0 and 59.

DATE

Returns the date in the form of a single number, for example
312.1992 being the 3rd December 1992.

TIME

Returns the time in the form of a single number, for example 1503.23 - being 3 minutes and 23 seconds past 3 in the afternoon.

ZEROTIME

Resets the elapsed time counter. Probably best included in an IF, THEN, ELSE entry.

TIMELAPSE

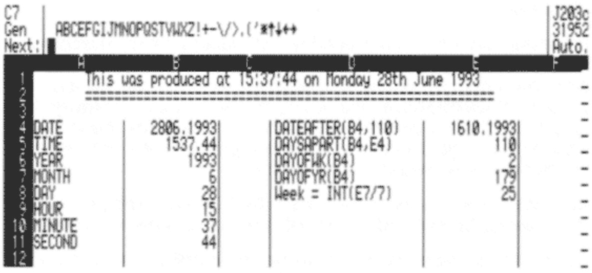
Returns the elapsed time since the ZEROTIME function was last operative. This is in seconds.

DELAY(*n*)

Does nothing until *n* seconds have elapsed. It may be a cell reference or a value. For practical reasons make it a cell reference with a value 0 until you actually want to run you application.

BLEEP

Sets off an audible tone. This can be used as a warning signal. On a sheet that takes a long time to calculate you could include the BLEEP function in the final cell to be calculated. You will then get a beep once re-calculation has finished.



```
C7     |                                                            |J203c
Gen    | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓↔                            |31952
Next:  |                                                            |Auto.
          A            B          C              D            E          F
 1             This was produced at 15:37:44 on Monday 28th June 1993
 2             ======================================================
 3
 4  DATE         |   2806.1993|     |DATEAFTER(B4,110)  |   1610.1993|
 5  TIME         |     1537.44|     |DAYSAPART(B4,E4)   |         110|
 6  YEAR         |        1993|     |DAYOFWK(B4)        |           2|
 7  MONTH        |           6|     |DAYOFYR(B4)        |         179|
 8  DAY          |          28|     |Week = INT(E7/7)   |          25|
 9  HOUR         |          15|     |                   |            |
10  MINUTE       |          37|     |                   |            |
11  SECOND       |          44|     |                   |            |
12               |            |     |                   |            |
```

Tutorial XIV

An introduction to 'command' functions

There are certain functions that we are now going to come across that are fundamentally different to those we have seen so far. The principle difference is that they act on a cell or cells other than the one in which they have been entered. They can be looked on really as more like commands than like the mathematical functions we have come to understand.

These functions are useful in that they can be used to automate some actions that you may normally have had to do yourself if you were using direct commands.

For example they can be used to create a worksheet that performs an entire set of calculations as soon as it has been loaded and the recalculation started. Certain of the demo files, IRR for example, use this feature.

They are also useful for creating subroutines that perform a range of actions or calculations automatically. We will see later how such a technique can be used to create and fill a table of values.

Examples of functions that behave rather like commands are **SET** and **INIT**, which assign a value to a distant cell rather than the one they are in, or **INC** and **DEC** which can alter the value held in a distant cell.

There are also some command-like functions that may read a value from a distant cell but do not act on anything in particular. An example would be one of the Graphics functions we will see later such as **MAINTITLE(*crd*)**. This reads the text to be displayed in a graph title from a distant cell, but it cannot display the answer it gets in its own cell because of different format types - it reads from a text cell but is used in a numeric cell.

Remember that because they are FUNCTIONS with arguments they must always be entered into a cell that has a numeric format.

Because the command functions act on a cell or cells other than the one in which they have been entered, and because that distant cell may sometimes contain text data, it is sometimes unclear what value the actual function cell itself will display. In fact, sometimes you will find they display the same value as held in the distant cell (if they can) and sometimes they will just display a zero.

To avoid confusing the display of the sheet you may wish to place these command functions in a seldom seen area of the sheet. It doesn't matter where they go - the Notebook spreadsheet will always find them.

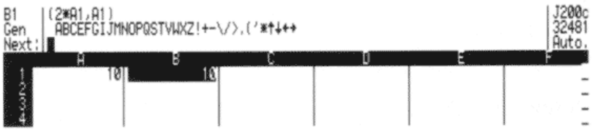
Multiple function lines and dividing commas

When you are entering functions into a cell it is valid to insert a dividing comma between them. The effect of a dividing comma is to cause an effective restart as if what follows was the beginning of the line.

The value that will be displayed in the cell will be that of the expression after the last dividing comma.

For example start with a blank sheet and enter **10** into A1 and this expression into B1:

2*A1,A1

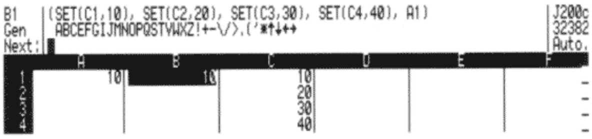


```
B1     |(2*A1,A1)                                                   |J200c
Gen    | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓↔                            |32481
Next:  |                                                            |Auto.
          A            B            C            D            E          F
 1             10|          10|            |            |            |
 2               |            |            |            |            |
 3               |            |            |            |            |
 4               |            |            |            |            |
```

The first calculation, 2*A1, is performed but is effectively forgotten - all that appears in B1 is a copy of A1 because this is what appears after the comma. This feature so far looks like a waste of time and memory but consider its application to the 'command' functions described above.

Try entering this expression into B1

SET(C1,10),SET(C2,20),SET(C3,30),SET(C4,40),A1



```
B1     |(SET(C1,10), SET(C2,20), SET(C3,30), SET(C4,40), A1)        |J200c
Gen    | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓↔                            |32382
Next:  |                                                            |Auto.
          A            B            C            D            E          F
 1             10|          10|          10|            |            |
 2               |            |          20|            |            |
 3               |            |          30|            |            |
 4               |            |          40|            |            |
```

Because the command functions all act on DISTANT cells their effect is performed on the sheet even though cell B1 finishes by only displaying the value obtained after the last comma. None of the earlier commands are wasted. This technique is also useful with the **BLEEP** function.

It should be obvious that only these command functions have a useful effect if they are followed by a comma.

Not all command-like functions can be treated in this way and some of the Graphics functions in particular will give unpredictable results if followed by a comma.

I/O FUNCTIONS

It is unlikely that these functions will be of any use but they are included for compatibility with other versions of the Cracker spreadsheet, should you wish to use files from the Notebook on other computers. Use of these functions, especially OUT, on the Notebook will almost inevitably lead to the machine crashing in which case all you can do is switch it on holding down [Function], [Stop] and [←Del] but this will lose all data you have stored in the Notebook.

IN(*port***)**

Reads an 8 bit port given by the number or cell reference *port.*

OUT(*port*, *value***)**

Outputs a *value* given by a number or cell reference to the *port* given by a number or cell reference.

Tutorial XV

Producing graphs and charts

The graphics part to the Notebook spreadsheet has been designed to allow you to create a graph with the minimum of work but you must of course provide the spreadsheet with some information about what you want plotted and how.

To do this there are some special functions, detailed below, that tell the graph plotter just what you want it to do. These can be entered anywhere you like in the sheet - the Notebook spreadsheet will be able to find them.

There are two distinct types of graph that may be plotted:

1) For the business chart types you just supply a single set of values (Y values) and each is plotted at a subsequent X position. You also supply a set of labels for the X axis which will often be some time interval (such as the months of a year). You can actually have several Y values for each point on the time axis.

2) For the scientific types of graph you supply (x, y) co-ordinate pairs for the points to be plotted. In this case the idea of "time labels" does not exist although, obviously, the X values could indeed be time increments.

PLOT1.MEM and PLOT2.MEM, included in the demonstration files, are examples of business and scientific graph plotting. You may wish to load these spreadsheets and eXchange the rule/formulae commands to see how the functions are used in practice.

To see how a graph is plotted, zap any sheet that is loaded and then type [D]**PLOT1**[↵] to load the demonstration file called PLOT1. The command to force the spreadsheet to draw a graph is T for Trace. Type [T][S][↵] to send the trace output to the screen. This will take some time; the rotating indicator shows that the spreadsheet is still working.

Initiially a small version of the graph is shown on the screen. This is scaled so that the complete picture fills the screen. You get an overall picture of how the output would look when printed. Press the [←] and [→] keys to switch between this reduced view and a larger view of the graph. When the larger view is shown the [↑] and [↓] keys may be used to scan up and down the picture.

When you have finished looking at the picture press [Stop] to return to the normal spreadsheet display. If you type [T] again you will see that you can also trace to printer. This offers the option of normal or high quality output. The printing of graphs from the spreadsheet will only work on Epson compatible printers. Specifically, the printer must support ESC K for normal output and ESC L for high quality output.

Graph functions

Here are the functions that are used to define what you want to plot. This is followed by a description of how they are actually used:

Note that where *crd* is specified in the functions below you must put it in the form shown. If you don't the current values may not be properly passed to the plotting section of the program. You can't use numerical or expression equivalents except where stated.

TYPEPLOT(*crd***)**

All charts must include this function before you try to trace them. This function is used to define the type of graph or chart plot you want. Give a value between 1 and 18 in the cell referred to by this function. If, for example, you want a Pie Chart use the function TYPEPLOT(A1) and put the value 6 into cell A1. Do not try to use TYPEPLOT(6).

Business graphs

Use TYPEPLOT values 1..6 as described below, (these graphs require TIMELABELs rather than actual XVALUEs, this is explained below)

1. Bar chart (histogram) - there may be more than one bar over each time label
2. Stacked bar chart - each bar for each time label is superimposed over the others so that only the overlap is shown
3. Line chart - lines join each point on each of the plot categories

4. Area chart - as above but the enclosed area is hatched.
5. Hi-lo chart - each of the plot categories are shown as a marker above each time label. The markers are joined by a vertical line.
6. Pie chart - the familiar segmented circular chart.

Statistical, Scientific and Engineering

Use TYPEPLOT values 7..18 as follows. For all these types of chart you must give x and y coordinate pairs using XVALUE and YVALUE.

7. X:Y line joining points
8. LogX:Y Line joining points
9. X:LogY Line joining points
10. LogX:LogY Line joining points
11. X:Y points only
12. LogX:Y points only
13. X:LogY points only
14. LogX:LogY points only
15. X:Y points and Line of Best Fit
16. LogX:Y points and Line of Best Fit
17. X:LogY points and Line Of Best Fit
18. LogX:LogY points and Line of Best Fit

MAINTITLE(*crd***)**

The cell co-ordinate is a pointer to a text cell where the main title is to be found. The use of MAINTITLE is optional. If you don't use MAINTITLE, the title area of the chart will just be left blank.

SUBTITLE(*crd***)**

The same idea as for the MAINTITLE applies to the SUBTITLE. This is printed directly below the main title in a slightly smaller font.

YTITLE(*crd*)

Where *crd* identifies a cell of text format holding the title to be written up the Y axis on the left hand side. YTITLE is not used for the Pie chart (TYPEPLOT 6). Use of YTITLE is optional.

XTITLE(*crd*)

Where co-ordinate *crd* refers to a text format cell where the X axis title is. The XTITLE is the one across the bottom of the graph. XTITLE is not used for the Pie chart (TYPEPLOT 6). Use of XTITLE is optional.

XLABEL(*crd...crd*), or XLABEL(*crd, crd, crd, etc.*)

This points to the text cells containing X labels which are the legend box items on the right of the chart explaining what the plot lines/bars etc. refer to. The spreadsheet also uses this function to discover how many plot lines/columns you want to display so you MUST include the XLABEL function.

Unlike the other functions in this list, the range can also be a list of individual items, each one pointing to one of the X labels.

Please make an effort to get the number of X labels correct as the program counts them to see how many lines or groups of items there are. If, for example, you get it wrong and enter too many Xlabels you will be informed there are Y values missing because it is expecting to have to draw further lines.

All types of chart *must* have an XLABEL function. It is not optional.

TIMELABEL(*crd...crd*)

The time labels are the tagged items on the X or bottom axis that show what exactly is being plotted on a business chart. They are not always time labels but it is very common in financial graphs for example to plot values according to month or year. This function is only used with plot types 1...6; the other graph types will expect actual numerical values on the lower line - given by the XVALUE function. For plot types 1..6 there MUST be a TIMELABEL function. If you don't actually want time labels, then just point the function to cells containing a single space character (they must not be completely blank).

YVALUE(*n, crd..crd*)

This function points to the actual Y values to be plotted. The '*n*' refers the number of the plotting line to which the values apply. There will be one of these functions for each line. All types of chart make use of the YVALUE function.

YMAXIMUM(*crd*)

The *crd* in this case can be a value or a reference to a value that specifies the maximum figure to be shown on the Y axis. Your choice will be rounded to a suitable nearby value to improve the presentation. Use of YMAXIMUM is optional. If it is not used the upper limit on the Y axis will be the same as the largest Y value.

YMINIMUM(*crd*)

This is a similar function to the above for a minimum value.

Note that because the Notebook spreadsheet rounds the value for the minimum to the best nearby value to give an attractive display, you may have to experiment and perhaps set the YMINIMUM value lower than you first thought in order to get the desired display.

XVALUE(*n, crd..crd*)

This function points to the actual X values to be plotted. The '*n*' refers the number of the line to which you are referring. There will be one of these functions for each line. XVALUE is only used for plot types 7...18, the TIMELABEL function is used for plot types 1...6

XMAXIMUM(*crd*)

The *crd* may be a value or a reference to a value that specifies the maximum value to be shown on the X axis. Your choice will be rounded to a suitable nearby value to improve the presentation. (Plot types 7...18 only)

XMINIMUM(*crd*)

Similarly for a minimum value. See the notes given above for YMINIMUM. The XMINIMUM command only works on plot types 7 or above.

Remember that because the graphics commands are entered as FUNCTIONS followed by values or co-ordinates each of the above should be placed into a cell that has been given a numerical format, even though the information they refer to may be text. The text itself of course has to be placed in cells of a text format.

You may be wondering what each of these cells that contain the above functions will actually display on the screen. In the majority of the cases the display will show zero. Some of the functions will display a numeric value if it is felt to be useful. For example the cell that contains the XLABELS function will show the number of Xlabels defined - that is, the number of sets of data to be plotted. The cell that contains TYPEPLOT will show the number of the graph type chosen.

Making a graph in practice

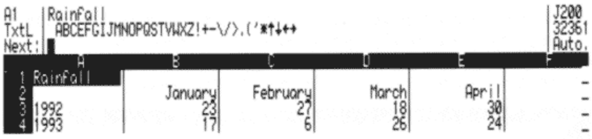
Business type

In a business type of graph you just have one or more sets of Y values that are to be plotted at discrete intervals along the X axis. You do not specify actual X values, the first point is plotted in column 1, the next in column 2 and so on. You can, however, define "time labels" which are text labels to be placed on each discrete X position. The following shows how you can make a business chart. Here is some sample data that we might want to plot:

Rainfall				
	Jan	Feb	Mar	April
1992	23	27	18	30
1993	17	6	26	24

Prepare a sheet with this data that you want to plot. The actual data values can be up and down columns or across lines, as long as they lie in a range of co-ordinates. One obvious way to set out this data would be exactly how we have written it above. All the labels (including 1992 and 1993) are entered into text format cells - start entry with " to use the global text format.

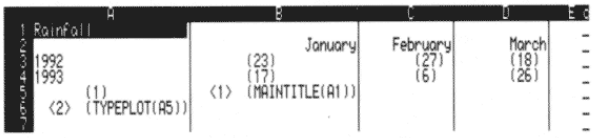
The numeric values are entered into general format cells using ⌨ to start entry:



```
A1   |Rainfall                                                    |J200
TxtL |  ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→                          |32361
Next:|█                                                           |Auto.
          A              B              C              D              E         F
 1 Rainfall
 2                    January|      February|         March|         April|       -
 3 1992                    23|            27|            18|            30|       -
 4 1993                    17|             6|            26|            24|       -
```

In addition to the actual data and labels we will need some functions to tell the spreadsheet what is to be plotted. It is not critical how the other items are laid out so just set aside an area to put in the plotting instructions. Each of these instructions is a pointer to where the particular data items you want displayed are to be found.

Start by putting the TYPEPLOT function into cell A6. Remember that you must include a cell reference in the TYPEPLOT function, such as **`TYPEPLOT(A5)`**. Don't just use TYPEPLOT(*number*). Once you have put in the function to define the type of chart, you can then put in pointers to the various titles, although these are optional. These are MAINTITLE, SUBTITLE, YTITLE and XTITLE. In this case, just enter **`MAINTITLE(A1)`** into cell B5. So far you may have something like the following:

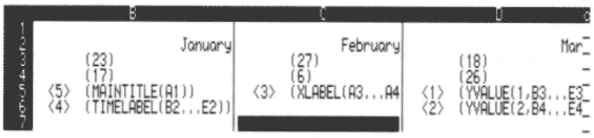


```
                   A                        B                   C               D      E G
 1 Rainfall
 2                                         January|        February|          March|   -
 3 1992                           (23)            |            (27)|           (18)|   -
 4 1993                           (17)            |             (6)|           (26)|   -
 5              (1)        <1>    (MAINTITLE(A1)) |                |               |   -
 6    <2>   (TYPEPLOT(A5))|                       |                |               |   -
 7                                                |                |               |   -
```

Next you must tell the spreadsheet which cells contain the time labels using the TIMELABEL function. Remember, these are typically the months or years that go across the bottom of the page, as in this example, but they don't in fact have to relate to time but usually do. In this case enter **`TIMELABEL(B2...E2)`** into cell B6.

For each position on the time axis we have two sets of data to be plotted, the values for 1992 and those for 1993. You must use the XLABEL function to give a name for each set of data that is to plotted. These labels will be used to print the key that appears to the right of the graph. The spreadsheet also knows that there are two sets of data to be plotted when you use this function. It cannot be omitted or the spreadsheet would not know how many sets of data there were. In cell C5 enter **`XLABEL(A3...A4)`**.

Lastly use the YVALUE function to point to the actual sets of data to be plotted. There will be one YVALUE entry for each set. In D5 and D6 enter **YVALUE(1,B3...E3)** and **YVALUE(2, B4...E4)**. You should have something that looks like the following:



```
          B                       C                       D            E
                   January               February                   Mar
 1
 2        (23)                    (27)                  (18)
 3        (17)                    (6)                   (26)
 4 <5>    (MAINTITLE(A1))    <3>  (XLABEL(A3...A4  <1>  (YVALUE(1,B3...E3
 5 <4>    (TIMELABEL(B2...E2))                     <2>  (YVALUE(2,B4...E4
 6
 7
```

If you particularly wish to specify the maximum or minimum values you want plotted then use the YMAXIMUM and YMINIMUM functions. Normally don't use these functions as the program will work out all the maximums and minimums for you automatically.

The only thing that remains is to actually instruct the spreadsheet to plot the graph. You do this by using the Trace graph to Screen (or Printer) command. Type T S ↵ to start the plotting process. Your current worksheet will be stored in the RESTART.MEM file and there will be a short delay while the spreadsheet works out how the graph is to be drawn. The screen will clear and a small version of the graph will be drawn to fit as best it can on the Notebook's 16 line screen. This will give you an overall feel for how the graph will look when printed.

You may want to see the picture in more detail and so it is possible to press the ← and → keys to get the spreadsheet to switch between this small version and an enlarged version of the picture. In the enlarged version you can then use the up and down arrow keys to move up and down the picture. Press the Space bar when you have finished looking at the on-screen chart. You will then be returned to the normal spreadsheet screen and your worksheet is reloaded from the RESTART.MEM file.

As mentioned above, the example file, PLOT1 contains a worksheet to produce a business plot. It is probably worth loading this file to see another example of the plotting instruction being used. Remember that you can use the X-eXchange command to switch the display to showing the formulas.

Try using the T S ↵ command on PLOT1 to see what it looks like. You could try changing the value held in cell A1, which is the type of plot. Use any number between 1 and 6 to see the various types of business chart available.

If you want to print a chart on your printer you use the command T P ↵. You are then asked whether you want to print it in normal or high quality. The former is quicker but the latter will give better results.

NOTE: The printing of graphs will only work on Epson compatible printers that support the ESC K and ESC L sequences for printing graphic data. If you are using a laser printer, see if it can be switched to emulate an Epson type of printer before attempting to print any graphs from the spreadsheet.

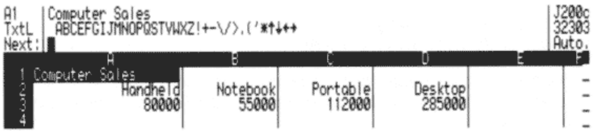
Pie charts

The first 5 types of business graph are all built in the same way. You have a TYPEPLOT to select the plot type. XLABEL identifies the labels for the "key" to the graph and also tells the spreadsheet how many lines/columns are to be drawn. The YVALUE function is used to tell the spreadsheet where each set of data is located. TIMELABEL gives the legends for the X axis. In addition to these you may also, optionally specify MAINTITLE, SUBTITLE, YTITLE, XTITLE, YMINIMUM, YMAXIMUM.

Pie charts are just a little different as the following example will show. In this case the aim is to show a set of data as fractional parts of a circle. Let us assume that we have the following figures to plot:

Computer Sales	
Handheld	80,000
Notebook	55,000
Portable	112,000
Desktop	285,000

Enter this data onto a blank worksheet as follows:

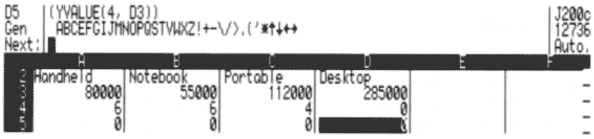


```
A1    |Computer Sales                                              |J200c
TxtL  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓↔                            |32303
Next: |                                                            |Auto.
              A            B            C            D            E       F
 1 Computer Sales
 2             Handheld     Notebook     Portable      Desktop
 3                80000        55000       112000       285000
 4
```

You must use TYPEPLOT to request plot type 6. Enter the value **6** into cell A4 and the **TYPEPLOT(A4)** function into B4. Each segment of the circle must have an XLABEL so in cell C4 enter the function **XLABEL(A2...D2)**. Although it is not used for anything other than a title on the chart, you must also have a TIMELABEL function. So put a **TIMELABEL(A1)** function into cell D4.

All that remains is to tell the spreadsheet where the data to plot is located. Each segment is a separate YVALUE so you must have separate YVALUE functions for each segment. In A5 enter **YVALUE(1,A3)**, in B5 enter **YVALUE(2,B3)**, then **YVALUE(3,C3)** and **YVALUE(4,D3)** in cells C5 and D5.

You should end up with something like this:



```
D5    |(YVALUE(4, D3))                                             |J200c
Gen   | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓↔                            |12736
Next: |                                                            |Auto.
         A            B            C            D            E         F
 2 Handheld    |Notebook    |Portable    |Desktop     |
 3        80000|       55000|      112000|      285000|
 4            6|           6|           4|           0|
 5            0|           0|           0|           0|
```

All that remains is for you to type the T S ↵ command to see the result. Remember that the ← and → keys may be used to switch between the full page and magnified view. In the magnified view you can scroll up and down the picture using the ↑ and ↓ keys.

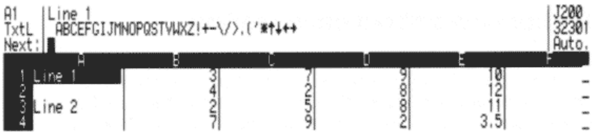
Scientific graphs

Unlike the business types of plot, the scientific charts require you to give (x,y) co-ordinate pairs for the points to be plotted. In this case you do not use the TIMELABEL function. Instead, you specify a set of XVALUEs in the same way as YVALUEs

An example will probably make this clear. Say, for example, that you want to produce a chart with two distinct lines on it. For each line you will give four points as follows:

Line 1:	(3,4), (7,2), (9,8), (10,12)
Line 2:	(2,7), (5,9), (8,2), (11,3.5)

Enter these data values onto a blank sheet as follows. The entries in cells A1 and A3 will be used as the XLABELS on the chart to identify the two lines.



```
A1    |Line 1                                                      |J200
TxtL  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓↔                            |32301
Next: |                                                            |Auto.
         A            B            C            D            E         F
 1 Line 1               3|           7|           9|          10|
 2                      4|           2|           8|          12|
 3 Line 2               2|           5|           8|          11|
 4                      7|           9|           2|         3.5|
```

As before you need to use the TYPEPLOT function to tell the spreadsheet how it should plot the data. For this example we will initially use type 7 so put the value **7** into cell A5 and the function **TYPEPLOT(A5)** into cell B5.

You must also tell the spreadsheet where it should find the Xlabels. Put the function **XLABEL(A1, A3)** into cell C5. As well as telling the spreadsheet where to find the labels, this has the secondary effect of telling it that there are two sets of data. The cell displays the value **2** to remind you how many sets of data the spreadsheet thinks there are.

All that remains is to let the spreadsheet know which cells contain the data points. Put the following functions into the following cells:

XVALUE(1,B1...E1)	in cell A6
YVALUE(1,B2...E2)	in cell B6
XVALUE(2,B3...E3)	in cell C6
YVALUE(2,B4...E4)	in cell D6

You may now use the T S ↵ command to see the result on the screen. It looks a bit sparse and you may like to add some of the following: MAINTITLE, SUBTITLE, XTITLE, YTITLE for a professional looking result.

Remember, if you would like to print a copy of the chart use an Epson compatible printer and give the command T P N ↵ or T P H ↵.

You can use these random sets of data points to see one further facility offered by the spreadsheet. Change the plot type number, in cell A5, to 15. This selects an X:Y chart with lines of best fit. This time, when you plot the chart, instead of the points just being joined, a straight line that best fits the given data points will be drawn. The key at the side of the graph will show the calculated function of the two lines in the form y=m*x + c. (The "*" is used to mean "times").

Further points about chart plotting

If you do any EDITING of the plotting instructions or make any changes that do not force a recalculation, you may end up with an error message or values that do not seem to be true. This is most likely when you use direct values in your functions rather than cell co-ordinates.

An example may be:

YMAXIMUM(20000)

and a case where it would not occur would be:

YMAXIMUM(B7)

Don't worry about this, just remember the safest way to handle plotting functions that do not contain cell references is to overwrite them rather than edit them.

The instructions for plotting are updated when a recalculation is done, so if you have made changes the latest instructions may not have been passed causing an error. If you get such an error message just use the ! force recalculation command and try the plot again.

Tutorial XVI - Automating Data Manipulation

Using macro command groups

Often you will want to go through the same set of commands repeatedly. A typical example is the changing of the format of all the cells in one column. This can be both time consuming and tedious. To get round this situation you should use the * macro command. A macro is a computing term that you will come across in many programs. You may already be familiar with macros in the word processor. A macro is just a sequence of key presses that you can easily recall to save you a lot of typing.

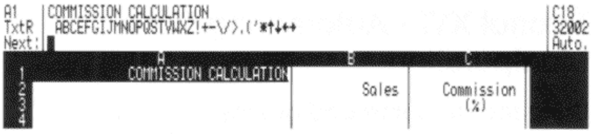
The Notebook spreadsheet will let you create predefined sets of linked commands. Just enter the command letters into a text format cell as if you were actually typing them in to be acted on immediately. Where you would want to use a ↵ enter the @ symbol instead.

To call the macro to perform these commands you only have to type * followed by the co-ordinate reference to the cell in which the macro is stored. So ***A1** calls the macro in cell A1. The maximum size of a macro is limited but you can put your macro in more than one cell by finishing with a reference to the continuation macro. For example you might finish your A1 macro with a reference *A2 to force it to continue with the commands entered in A2.

If you want your macro to loop and be carried out repeatedly then finish it with a reference to itself. For example with cell A1 this would mean finishing with *A1. Don't worry about this causing an endless loop, there are lots of ways of making the macro come to an end.

As an illustration the following example takes a column of numbers in financial format and changes them all to one place decimal format.

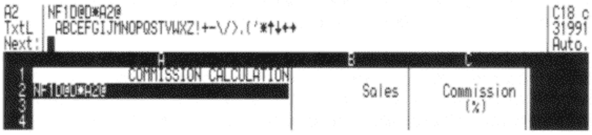
Load the demonstration file called SALES.MEM. If you have just started the spreadsheet type the following - □**SALES**↵.



```
A1    |COMMISSION CALCULATION                          |C18
TxtR  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→              |32002
Next: |▌                                               |Auto.
              A                      B          C
1        COMMISSION CALCULATION
2                                  Sales    Commission
3                                               (%)
4
```

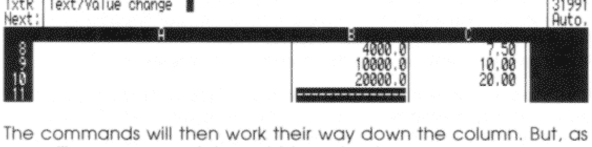
Start by moving the cursor to cell A2 where the macro is to be entered. Type " to get into text entry mode and then type **NF1D@D*A2@** which says 'New Format one place Decimal, carriage return, cursor Down and finally do macro A2 again'.

The @ symbol stands for the ↵ carriage return.



```
A2    |NF1D@D*A2@                                      |C18 c
TxtL  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→              |31991
Next: |▌                                               |Auto.
              A                      B          C
1        COMMISSION CALCULATION
2  NF1D@D*A2@                      Sales    Commission
3                                               (%)
4
```

Next move the cursor to the location you want the macro to start its operations. In this case it is cell B5 so type □**B5** to move quickly to that cell. Now type ***A2**↵ to start the macro going.



```
B11   |---------------                                 |C18 c
TxtR  |Text/Value change  ▌                            |31991
Next: |                                                |Auto.
              A                      B          C
8                                   4000.0        7.50
9                                  10000.0       10.00
10                                 20000.0       20.00
11                             ---------------
```

The commands will then work their way down the column. But, as you will see, a macro is brought to a stop by any error message.

You can see that the macro has come to a stop on cell B11 because it is not possible to change a text format cell to a numeric format. You see the message **Text/Value change**. Just type Stop and your looping macro operations will be completed and you will be in a position to go onto your next command.

A macro differs from a subroutine function, which you will learn about next, in that it is a list of direct commands rather than functions - the macro can contain a command such as NF for New Format, which is something that cannot normally be entered into a cell as part of an expression. The subroutine, on the other hand, allows one that performs normal expression functions repeatedly.

It is of incidental interest that if you were to restrict your macro definitions to the first 9 cells of column A they could be accessed by just typing *1,*2 etc. Move to cell C5 (type □**C5**↵) and then type ***2**↵ to see this in operation. This time the macro comes to a stop when an attempt is made to change the format of a blank cell - the NF command only works on cells that already contain data.

Functions that allow looping

Sometimes it would be very useful if you could use a few formulae repetitively to work towards an answer. For instance when working out the internal rate of return example given in the demonstration files (the file called IRR.MEM), a short entry has been set up that tries a range of possibilities and stops at the nearest.

The functions that allow you to do these repetitive calculations are often referred to as loops. The Notebook spreadsheet provides you with two functions, **DO** and **WHILE**, designed to make the setting up of loops very easy.

Unlike the IF, THEN, ELSE example seen earlier the command word DO is not followed directly by an expression but rather by a reference to a range of cells that contains the desired expression. This saves you a lot of typing if your calculation is long and complex.

Note that you must specify a RANGE, so your working must be in one line or one column and not in a block. However, the function looks at the range and finds the highest and lowest order of recalculation number and then all those entries between the numbers are recalculated no matter where they are in the sheet. It does not matter if some of your intermediate calculation numbers are not in the specified range as they will still be correctly calculated.

To use a loop enter the **DO(*range*)** function which performs the desired calculation followed by a comma then any other expression or function that you wish. Usually this second function will change a value somewhere that acts as a loop counter (counts the number of times that the calculation has been done).

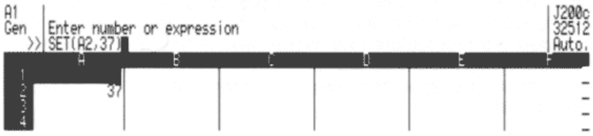
There are several related functions that make setting up a counter easy for you namely **INIT**, **SET**, **INC** and **DEC**. These respectively initialise a cell entry, assign a value to a cell entry, increment a value (increase by one) and decrement a value (decrease by one).

After the loop counter section you should type in a further comma and then use the **WHILE** function. This function has a logical argument such as a logical expression finding whether the loop counter has reached a certain value, often zero. If the WHILE function is TRUE then the cell formula is started again at the DO and repeated to the WHILE until the WHILE becomes FALSE.

That is a basic description and it will seem fairly complicated at this stage. Do not worry about it yet, follow through the examples and then go back and look at this section again.

First, you are going to look in more detail at how the **SET** function works. It is of the form **SET(***crd, value***)**. The value can also be an expression and can also include the referenced co-ordinate. SET(A2, A2+1) is valid and works in exactly the same way as INC(A2). The SET command works in a similar way to entering a constant value into a cell using the entry command, except that the value is entered as a result of the expression rather than directly by you. It is therefore a useful way of automating the entry of cell constants and ensuring that the value will be updated if the referred cell values change.

If you think about it, SET is a way of filling one cell with a value whilst you are entering an expression into another cell. Try entering the function **SET(A2,37)** into cell A1 of a blank sheet. Don't press ↵.

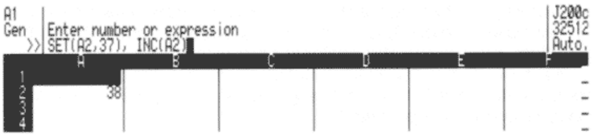


```
A1   |                                              |J200c
Gen  |Enter number or expression                    |32512
   >>|SET(A2,37)                                    |Auto.
            A          B          C          D          E          F
1
2          37
3
4
```

You may note that the destination cell is set even before the expression is fully entered.

If you decided to press ↵ at this stage, to end the expression entry, the actual cell that contains the SET function (A1) would itself take on the value shown in the distant cell that has been set. This is primarily because it hasn't been shown anything else to display rather than for any logical reason.

Next see how the **INC** increment function works. This is not a particularly useful expression by itself but you will see later how it can form an important part of a DO WHILE loop. After the SET function that you have already typed continue with **,INC(A2)**. Once again, the function works even before the expression is completed, so as you enter the formula you will see A2 first had the value 37 and then 38 as it is incremented.



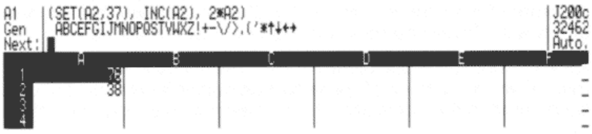
```
A1   |                                              |J200c
Gen  |Enter number or expression                    |32512
   >>|SET(A2,37), INC(A2)                           |Auto.
            A          B          C          D          E          F
1
2          38
3
4
```

Now although you are manipulating the values that are held in cell A2 don't forget that your current cursor position is still A1 and that it is into A1 that you are entering your expression.

Remember that A1, i.e. the current cursor position, will take on a value that is calculated from the expression or function after the last dividing comma in your expression list.

You can therefore put a further comma in and then an expression or value that doesn't manipulate A2 such as **2*A2**↵. You will see that A2 now contains 38 and A1 shows the value 76 which is 2*A2.

The effect of dividing commas is to cause an effective restart as if what follows was at the beginning of the entry line. You should understand that only functions like SET, INC and DEC have any actual effect on the worksheet if there is a later dividing comma, i.e. they act on cells other than the current one, so a restart does not alter the values that these cells have assumed:



```
A1   |(SET(A2,37), INC(A2), 2*A2)                   |J200c
Gen  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→            |32462
Next:|                                              |Auto.
            A          B          C          D          E          F
1          76
2          38
3
4
```

In this case the entry at A1 has ended up with the value 2*A2 which is 76. The cell entries as a whole have become a series of instructions executed in turn.

Subroutines using the DO function

As well as being a part of a loop the DO function can be used without the WHILE to act as a call to a subroutine that is only executed once every time it is called.

You may be familiar with subroutines but if not, here is an explanation.

A subroutine is a group of formulae which you may want to use repeatedly. Rather than enter the formulae many times over you simply have to type them in once and access them with the DO function. You will see the similarity here with a macro command. Every time the subroutine function is found in a cell during recalculation the whole of the group is calculated again. Now have a look at it in practice:

Start with a blank sheet and enter the formula **2*A4** into cell A3. A line of question marks appears because the formula in A3 is waiting for you to input a value into A4. In cell A1 enter the following: **SET(A4,5), DO(A3...A4)**↵. When you type the closing parenthesis of the set command the value 5 appears in cell A4 but the result (in cell A3) does not appear until you press ↵

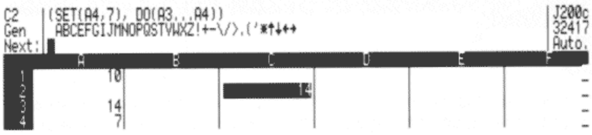
The same function (2*A3) could be used several times to double values from all over the sheet. Each time the "subroutine" is called you just pre-load A4 with the parameter to the function and then use DO to execute the function in A3.

Note that although you only want the function in A3 to be recalculated, the argument of the DO function MUST be a range; it cannot be individual entries (although it can be just two, one of which is blank) or a block. However, this is not as restrictive as it may sound because the DO function will calculate a complete sequence of formulae. Each cell in the range you specify will have an order of calculation number. The DO function will find the one that has the lowest number and the one that has the highest.

As each function is calculated in turn, every cell in the sheet that falls within the bounds of these two calculation numbers will also be automatically recalculated. This means you will get the effect you want provided your range includes the start point you want and the end point you want.

When you call a subroutine like this you can make sure the result is copied into the calling cell by following the expression by a comma and a reference to the cell that has the answer, in other words by effectively resetting the expression you have typed and by copying the value of the answer cell into the current cell.

Now you can try using the subroutine group we have set up to multiply 7 by 2. To illustrate that you can call the subroutine from anywhere in the sheet you can do this in cell C2: Enter **SET(A4,7),DO(A3...A4),A3** into cell C2.



```
C2   |(SET(A4,7), DO(A3...A4))                      |J200c
Gen  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→            |32417
Next:|                                              |Auto.
            A          B          C          D          E          F
1          10
2                                14
3          14
4           7
```

Note that the cells that contain the subroutine are also changed as well. Thus you end up with 14 in cell A3 as well as in C2. Because of this, it is as well to keep the subroutine cells well out of the way of the rest of the sheet.

The group used for subroutine calculation should not be referred to by the main body of the worksheet other than as a subroutine call or it can give nonsense values. You are, after all, changing the constants in it several times during the calculation.

You can change the cell values with SET commands before each DO function.

This is a trivial example but the same technique can be used for much larger sections of the worksheet.

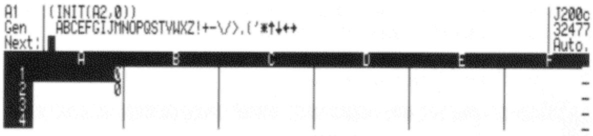
Looping using functions DO and WHILE

Now we are going to look at how to use the DO and WHILE functions in conjunction in order to build up a repeating loop.

Your first job when setting up a loop is to set aside a cell to act as a counter to keep track on how many times the loop has been performed. Do this with the **INIT** function.

The **INIT** function is almost identical to SET but the target cell will be given a very low order of calculation number and hence you can be sure it will be re-calculated early on in every forced recalculation. This is important because every time you run the loop INIT makes sure that the counter is reset to its original value. Using SET in these circumstances may mean that the counter cell increases in value after the loop is used once and the condition tested for by the WHILE function may never be met.

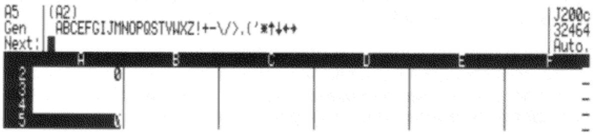
Blank the sheet and then put the function **`INIT(A2,0)`** into cell A1:



```
A1   |(INIT(A2,0))                          |J200c
Gen  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓++     |32477
Next:|                                      |Auto.
        A       B       C       D       E       F
1       0
2       0
3
4
```

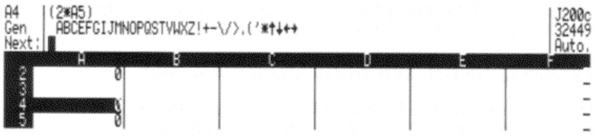
In this case the counter cell is A2. The object of this example is to multiply the value of the counter by 2 and then increment the value and do it again. A criterion for stopping will be specified.

Here a reference to the counter is placed in A5:



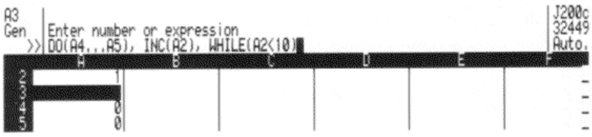
```
A5   |(A2)                                  |J200c
Gen  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓++     |32464
Next:|                                      |Auto.
        A       B       C       D       E       F
2       0
3
4
5       0
```

At this stage it has the value 0. The expression **`2*A5`** is entered into cell A4:



```
A4   |(2*A5)                                |J200c
Gen  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓++     |32449
Next:|                                      |Auto.
        A       B       C       D       E       F
2       0
3
4       0
5       0
```

The counter and formula are now set up so the loop can be established: In cell A3 enter **`DO(A4...A5), INC(A2). WHILE(A2<10)`**. This is just like the previous example of DO but now we have added an extra WHILE function to the end.



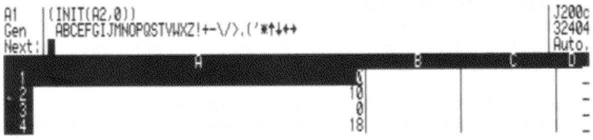
```
A3   |                                      |J200c
Gen  |Enter number or expression            |32449
   >>|DO(A4...A5), INC(A2), WHILE(A2<10)    |Auto.
        A       B       C       D       E       F
2       1
3
4       0
5       0
```

Each time you press **!** to force a recalculation. You will see A5 count from 0 to 9 and A4 (the result) will count up in steps of two.

The heart of the loop is the WHILE function which has the special property that if the logical argument that follows it is TRUE then the expression is wound back to the DO and repeated. In the example it remains TRUE while A2 is less than 10. But note that after the DO function the counter A2 is incremented. The multiply by 2 formula therefore has a different start value. Overall the effect is that on the screen you see A5 going from 0 to 9 while A4 goes from 0 to 18. Not a lot of use in this case but it shows the loop working 10 times. Later you will see the looping used to fill a table.

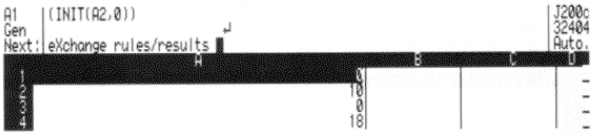
Reminder:

To make it easier to see the rules of the spreadsheet you can convert the display to show just the formulae. Let us try this now. First the column width needs to be increased, type N W 45↵

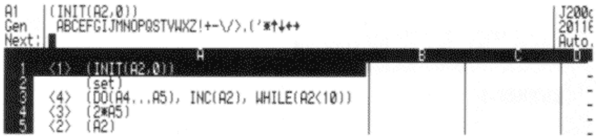


```
A1   |(INIT(A2,0))                          |J200c
Gen  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓++     |32404
Next:|                                      |Auto.
                    A                   B       C       D
1                                    0
2                                   10
3                                    0
4                                   18
```

Then the X for eXchange command is used:



```
A1   |(INIT(A2,0))                          |J200c
Gen  |                      ↵               |32404
Next:|eXchange rules/results                |Auto.
                    A                   B       C       D
1                                    0
2                                   10
3                                    0
4                                   18
```



```
A1   |(INIT(A2,0))                          |J200c
Gen  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓++     |2011
Next:|                                      |Auto.
                    A                   B       C       D
1   <1>  (INIT(A2,0))
2        (set)
3   <4>  (DO(A4...A5), INC(A2), WHILE(A2<10))
4   <3>  (2*A5)
5   <2>  (A2)
```

You can now see all the formulae. The number in front of the formula is the order of recalculation number.

The way the looping is organised means that the DO formulae will always be calculated once before the WHILE is tested. This ties in with the way DO and WHILE work in the main computer languages. If this is not the way you would like the loop to work you can precede it with an ordinary IF THEN logic test.

Note: It is NOT possible to have a DO-WHILE loop within a DO-WHILE loop.

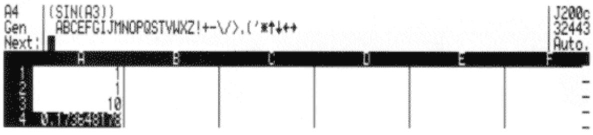
Table filling using the DO function

Because it provides an automatic method for filling cells with values, the SET command is useful for creating tables within the sheet.

You are going to see how to fill a table with the values of SIN(*x*) between 10 and 90 degrees. As before, you start by initialising a counter to keep track of where you have got to in the table. In this case cell A2 is set up with 1. Blank the sheet and enter **`INIT(A2,1)`** into cell A1

As you want the sine values every 10 degrees you can use the counter multiplied by 10. This formula is inserted in cell A3 as **`10*A2`**

Now you have a value for degrees you can refer to it with a SIN function in cell A4. Enter the formula **`SIN(A3)`** into that cell.



```
A4   |(SIN(A3))                             |J200c
Gen  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓++     |32443
Next:|                                      |Auto.
        A       B       C       D       E       F
1       1
2       1
3      10
4 0.173648178
```

The object is to fill a table in column B with the sine values at every 10 degrees. Somehow you must transmit the calculated value to the required slot. Once more you can use the SET function but together with the **CRD(***col, line***)** function rather than a specific co-ordinate defined when you first enter the function.

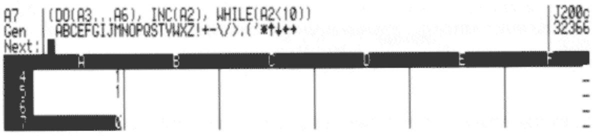
CRD is another special function that we have not discussed before. It can be used anywhere you would use a co-ordinate reference if typing directly. The two arguments are calculated to be a column and row number which are combined to make a cell reference.

In A5 enter the expression **`A4, SET(CRD(2,A2),A4)`**. This expression starts with **`A4,`** to ensure that it is not calculated until the result of A4 is known. It will get an order of calculation number higher than that for cell A4.

The CRD function is of the form **CRD(***column, line***)**. In this case the column number is 2 which is column B. The line number is the value of A2 which is 1 at present. The CRD function will therefore become the cell reference in B1 and will behave exactly as if the entry read SET(B1, A4). As you can see the cell B1 has already been set to the value of A4.

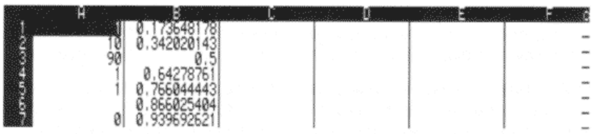
A2 is a counter and can be made to change in value by using a DO function. As A2 changes so will the CRD function give different cell references. In this way we can fill a column of cells with values and form a table.

The next step is for you to enter the DO - WHILE loop. In A7 enter **`DO(A3..A6), INC(A2), WHILE(A2<10)`**. A6 has deliberately been left blank for the moment. The sheet will then look like:



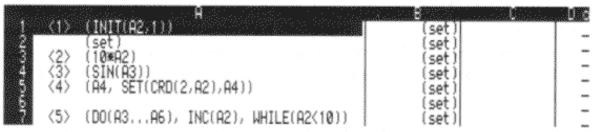
```
A7   |(DO(A3...A6), INC(A2), WHILE(A2<10))  |J200c
Gen  | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓++     |32366
Next:|                                      |
        A       B       C       D       E       F
4       1
5       1
6
7       0
```

The DO..WHILE says 'calculate the sine values as set out in cells A3...A5 then add one to the counter and do it again as long as the counter is less than 10 - that is between 1 and 9': Press **!** to force a recalculation and you will see column B filled with the values of SIN at 10 degree intervals again:



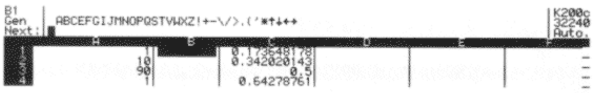
```
        A            B           C        D        E      F G
1              1| 0.173648178|
2             10| 0.342020143|                              -
3             90|         0.5|                              -
4              1|  0.64278761|                              -
5              1| 0.766044443|                              -
6               | 0.866025404|                              -
7              0| 0.939692621|                              -
```

Press X to change from the display of results to display the formulae used:



```
                       A                        B        C   D G
1  <1>  (INIT(A2,1))                          (set)|
2       (set)                                 (set)|          -
3  <2>  (10*A2)                               (set)|          -
4  <3>  (SIN(A3))                             (set)|          -
5  <4>  (A4, SET(CRD(2,A2),A4))               (set)|          -
6                                             (set)|          -
7  <5>  (DO(A3...A6), INC(A2), WHILE(A2<10))  (set)|          -
```

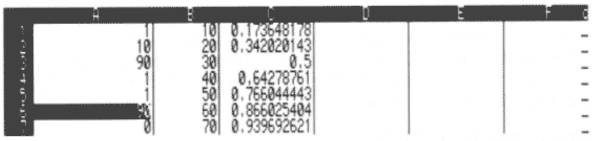
You can add to the sophistication of the display by actually having the degrees presented as well. To do this you must first insert another column. With the cell cursor on any cell in column B type ' C ↵ 8 ↵ ↵ G ↵. This inserts a new column B that is 8 characters wide and has a default column format of General.



```
B1    | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→                 |K200c
Gen   |                                                   |32240
Next: |                                                   |Auto.
      A          B          C            D        E       F
1              1|        | 0.173648178|
2             10|        | 0.342020143|                     -
3             90|        |         0.5|                     -
4              1|        |  0.64278761|                     -
```

With the new column B inserted you can see that the CRD function in cell A5 needs adjusting to refer to column C. This is NOT done automatically for you and so you will have to re-enter it or use the Edit command to change the first 2 to a 3. For the practice let us do the latter. Type J **A5** to move quickly to that cell and then type E ↵ to edit it. Move the cue to the right of the first 2 in the line and type ⌫ to remove it. Now type **3**↵. As soon as you leave the editor a recalculation is performed and this time column C is filled with the SIN values.

All that remains is to add another SET function to fill up column B with the degree values which are calculated in cell A3: This must be in the range of cells executed by the DO function so we will use cell A6 which was left blank. In that cell enter **`A4, SET(CRD(2,A2),A3)`**. Once again, the actual SET command is preceded by a reference to cell A4 so it is not executed until that cell has been calculated. You could now press **!** - this will force a recalculation to give the tables you want:



```
       A          B          C            D        E      F G
1             1|        10| 0.173648178|
2            10|        20| 0.342020143|                    -
3            90|        30|         0.5|                    -
4             1|        40|  0.64278761|                    -
5             1|        50| 0.766044443|                    -
6            90|        60| 0.866025404|                    -
7             0|        70| 0.939692621|                    -
```

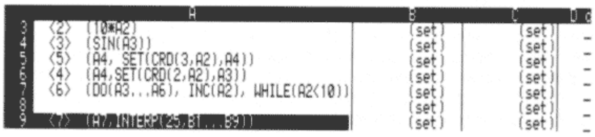
The SET command has one drawback compared to entering all of the values manually. The Notebook spreadsheet is not always able to allocate order of recalculation numbers to the cells that it itself has set. The next example will illustrate how this can cause some minor problems.

You are going to use the tables in columns B and C to interpolate a value for 25 degrees. You should enter the **INTERP** function in the following form to get the value you want. In cell A9 enter **`INTERP(25, B1...B9)`**↵

But this does not work as an error message about an ambiguity comes up. This is because you are trying to operate on a calculated table and the spreadsheet cannot resolve just exactly what calculation number to give your new entry relative to the table:

You can solve this problem by first putting a reference to the cell that created the table. In this case it was the DO function cell, so you have to put A7 before your INTERP function. That is: **`A7, INTERP(25,B1...B9)`**↵. It may not be logically obvious but this is a way of telling the spreadsheet that your new function should have a calculation number greater than that of the DO function. It was for this reason that you had to include the reference to A4 before the two SET commands in cells A5 and A6.

Switch to the display of formula with the X command and note the DO function has a calculation number of 6 and the INTERP follows it with 7 so the attempt to interpolate is only made once the DO function has finished and the ambiguity no longer exists.



```
                        A                          B         C     D G
3  <2>  (10*A2)                                  (set)|    (set)|
4  <3>  (SIN(A3))                                (set)|    (set)|  -
5  <5>  (A4, SET(CRD(3,A2),A4))                  (set)|    (set)|  -
6  <4>  (A4,SET(CRD(2,A2),A3))                   (set)|    (set)|  -
7  <6>  (DO(A3...A6), INC(A2), WHILE(A2<10))     (set)|    (set)|  -
8                                                (set)|    (set)|  -
9  <7>  (A7,INTERP(25,B1...B9))                  (set)|    (set)|  -
```

If you use the X command you can switch the display back to see the result of the interpolation which gives value exactly half way between the sine of 20 and the sine of 30 as you would expect. This is just a linear interpolation. The actual value of SIN(25) is 0.42262 but the INTERP value is close.

Limitations of CRD()

The CRD function should only be used with INIT, SET, INC and DEC to specify the cell to be acted upon. It cannot be substituted in every instance where you would normally specify a cell co-ordinate directly - it is only a pointer, not a receiver of data.

It can not be used to give the value of a cell if used in other expressions. It cannot be used, for example, in an expression such as 2+CRD(1,2) instead of 2+A2.

There is an automatic function that will do this which is the **VAL** or value function which returns the current value of the cell to which it refers. The arguments are formed in the same way as for the CRD function. So, for example, if A2 contains 3 the expression 2+VAL(1,2) will give the answer 5.

Iterative solutions - using circular references

As mentioned in the opening sections of this manual most spreadsheets will get very confused if you try to enter formulae that refer to each other in a circular way e.g. making A1 equal to 2*B1 and B1 equal to 3*A1.

The Notebook spreadsheet is more broadminded about such things, indeed not only will it let you set up such circular references but, if correctly done, they can be a powerful problem solving tool.

By setting up circular references to problems that are resolvable you will see that the spreadsheet makes an estimate about the values that belong in the cells in question.

If you then force repeated recalculations then this estimate moves closer and closer to the correct solution until an answer is found. This is known as an iterative method of problem solving.

Such repeated calculations can profitably be automated using a DO WHILE function.

Consider the next practical example.

Solving simultaneous equations

Simultaneous equations can be solved easily using the Notebook spreadsheet. The technique applies equally to awkward equations with the unknown you are seeking on both sides of the equation.

The principle is quite simple. You prepare a set of formulae for the answers as if all the other variables have known values. They don't because each formula depends on the results from the others. These are known as circular references.

When the last formula has been entered you force a recalculation and each formula uses the latest results available. This improves the result that each calculates. Further recalculations bring the calculated results nearer to the true results. This is a powerful technique that is often used for programmed solutions to equations. The difference is the formulae you enter are simple in comparison and you are fully in control of the solution. You can even automate the solution using the DO WHILE functions.

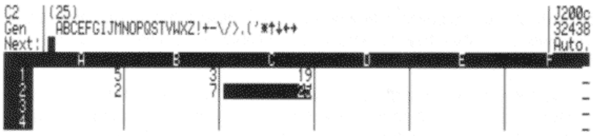
Here is an example for two equations and two unknowns. It is trivial but that makes it easier to see the principles.

The equations are:

5x + 3y = 19

2x + 7y = 25

You enter them onto the sheet using the constant multipliers of x and y and ignore the values of x and y themselves, i.e. in the form:

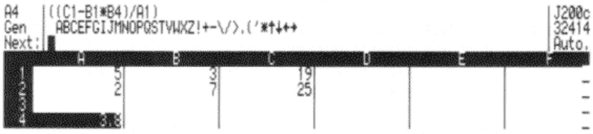


```
C2    |(25)                                           |J200c
Gen   | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→             |32438
Next: |                                               |Auto.
         A         B         C         D         E         F
1        5         3        19
2        2         7        25
3
4
```

Now you enter the solution formulae. Here you assume that the final solution for x will be in A4 and the final solution for y will be in B4. You can therefore refer to these cells as if they hold the answers you are seeking. First use the calculation to get x in terms of y. This is equivalent to manually rearranging the first equation with x on the left and everything else on the right.

x = (19 - 3y)/5

Insert this in A4 as **(C1-B1*B4)/A1**.

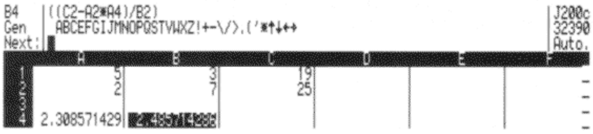


```
A4    |((C1-B1*B4)/A1)                                |J200c
Gen   | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→             |32414
Next: |                                               |Auto.
         A         B         C         D         E         F
1        5         3        19
2        2         7        25
3
4      3.8
```

B4 or y is unknown but refer to it nonetheless. As B4 does not yet contain anything it is assumed to be zero so this produces a first approximation for x of 3.8. Next you rearrange the second line in terms of the y unknown.

y = (25 - 2x)/7

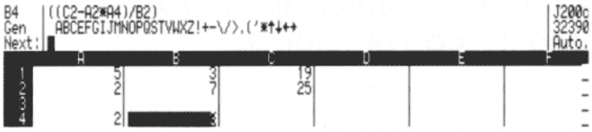
Insert this as **(C2-A2*A4)/B2** in cell B4.



```
B4    |((C2-A2*A4)/B2)                                |J200c
Gen   | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→             |32390
Next: |                                               |Auto.
         A         B         C         D         E         F
1        5         3        19
2        2         7        25
3
4 2.308571429 2.485714286
```

Note that A4 or x is used in this equation even though it doesn't have a true value. The first estimate of 3.8 is used. So these two equations now rely on the answer from each other.

Before you go any further, save the sheet as it is by typing C A F **temp**↵. Next you only have to press the **!** key to force a recalculation a few times and after about twelve recalculations the answers will settle down to the true answers of 2 and 3.



```
B4    |((C2-A2*A4)/B2)                                |J200c
Gen   | ABCEFGIJMNOPQSTVWXZ!+-\/>.('*↑↓←→             |32390
Next: |                                               |Auto.
         A         B         C         D         E         F
1        5         3        19
2        2         7        25
3
4        2         3
```

You may have found it tedious to have to repeatedly force a recalculation. This can be automated. Zap the current sheet and at the opening menu select Copy to reload the file called TEMP. Now enter the following into cell D4:

INIT(E4,1),DO(A4...B4),INC(E4),WHILE(E4<20)

This would force 20 iterations to be calculated. As you press ↵ the loop starts to operate and as cell E4 counts up towards 20 you will see the values in A4 and B4 repeatedly change until they settle at the final values of 2 and 3.